muSRFlow User Manual

Release 1.0.0-rc.4

Zaher Salman

Oct 10, 2026

Contents

1 Overview **3**
- 1.1 What you get . . . 3
- 1.2 Under the hood . . . 4
- 1.3 Two editions . . . 4
- 1.4 A first fit . . . 4

2 Getting started **5**
- 2.1 Prerequisites . . . 5
- 2.2 Desktop installation . . . 5
- 2.3 Running the desktop app . . . 6
- 2.4 Opening the web edition . . . 6
- 2.5 The Quick Start guide . . . 6
- 2.6 Support . . . 6

3 Runs and fitting **7**
- 3.1 Run selection . . . 7
- 3.2 Fit range . . . 8
- 3.3 Fit model . . . 8
- 3.4 Fit options . . . 9
- 3.5 Series fit actions . . . 9
- 3.6 View settings . . . 10
- 3.7 Fit and Plot . . . 10

4 Parameters and parameter plots **13**
- 4.1 The parameter table . . . 13
- 4.2 Selecting parameters for plotting . . . 14
- 4.3 The Choose X axis dialog . . . 14
- 4.4 The parameter-plot window . . . 14

5 Constraints **19**
- 5.1 Custom parameters . . . 19
- 5.2 Constraints . . . 19
- 5.3 Example: couple the magnetic field to the Larmor frequency . . . 20

5.4 Preview . . . 20

6 t0 / background **21**
6.1 Multi-group asymmetry fits . . . 21

7 MSR editor **23**
7.1 Ownership: GUI or editor . . . 23
7.2 Workflow . . . 23
7.3 File handling . . . 24

8 Results and diagnostics **25**
8.1 Result files . . . 25
8.2 Execution details . . . 25
8.3 Command log . . . 26
8.4 The status bar . . . 26
8.5 Run log window . . . 26
8.6 Outcome dialog . . . 26

9 Configuration sharing **27**
9.1 What is shared . . . 27
9.2 On the web . . . 27
9.3 On the desktop . . . 28
9.4 Notes . . . 28

10 Preferences **29**

11 Web and desktop editions **31**

12 Keyboard shortcuts **33**
12.1 Parameter selection (Parameters tab) . . . 33
12.2 Parameter-plot window . . . 33
12.3 MSR editor . . . 33
12.4 General notes . . . 34

13 Troubleshooting **35**
13.1 Fit or plot will not start . . . 35
13.2 “Nothing to fit” . . . 35
13.3 Fitted parameters look wrong or unchanged . . . 35
13.4 Parameter plot is empty or missing data . . . 36
13.5 Parameter plot hangs on “Preparing plot files” . . . 36
13.6 Web page cannot load data . . . 36
13.7 Restored configuration differs from the original machine . . . 36
13.8 Some links start with a different prefix . . . 36
13.9 Reporting problems . . . 37

14 Glossary **39**

Index **41**

muSRFlow is a graphical workflow for creating, editing, fitting, plotting, and inspecting `musrfit` MSR files. It is available as an Electron desktop application and as a browser application, and both editions wrap the same set of command-line tools (`musrfit`, `musrview`, `msr2data`, and `dump_header`).

These pages describe the end-user workflow. They are organised around the main window tabs, mirroring the Quick Start help that is built into the application (the `?` button).

CHAPTER 1

Overview

muSRFlow provides a modern interface over the classic μSR analysis tools. A typical session selects the data, chooses a fit model, fits it, inspects the result, and finally inspects how the fitted parameters vary run by run.

1.1 What you get

- A single window with the full workflow in tabs:
  - **Runs** – run selection, factorial fit range, fit model, fit options, and the Plot / FFT / RRF view settings.
  - **Parameters** – the parameter table with starting values, steps, limits, and plot selection.
  - **Constraints** – custom parameters and parameter constraints.
  - **t0 / Bg** – expert timing and background controls.
  - **MSR Editor** – the generated MSR document, with GUI-to-MSR and MSR-to-GUI synchronisation.
  - **Results** – generated files, execution details, and the full command log.
- Two one-click workflow actions in the main toolbar:
  - **Fit** – generate the MSR document and run the fit.
  - **Plot** – generate the MSR document and display the spectra.
- Interactive parameter plots that show how every fitted parameter evolves across the selected runs.
- Configuration sharing: the current analysis settings can be encoded into a link and restored on another machine.

1.2 Under the hood

muSRFlow does not reimplement the physics. It writes MSR files that are fed to the musrfit package, and renders spectra with `musrview`:

- `musrfit` – the muon-spectroscopy fitting engine. muSRFlow generates an MSR file from the GUI and runs `musrfit` on it. The fitted values (a `par.dat` parameter table) feed the parameter plots.
- `musrview` – the reference data viewer. On the desktop it opens its own native window; in the browser edition the server produces a ROOT canvas that is drawn with JSROOT inside the page.
- `msr2data` – converts a fitted MSR file into a parameter table (`par.dat`) that the parameter plot can read.
- `dump_header` – reads run metadata used to resolve portable run names and to guess initial parameters.

1.3 Two editions

- **Desktop (Electron)** runs the tools directly on your machine, manages its own data and working directories (set under **File / Preferences**), and can open and save MSR files from the File menu.
- **Web** runs the same interface in a browser. The data tree and the command-line tools live on the server, so file selection is limited to what the server exposes, and there is no local Preferences dialog.

Both editions share the same workflow and produce the same MSR documents. The differences that matter day to day are collected in *Web and desktop editions*.

1.4 A first fit

1. In the **Runs** tab, choose the instrument and year and enter one or more run numbers.
2. Adjust the fit range and the fit model as needed.
3. Press **Fit**.
4. Open the **Parameters** tab to verify and adjust starting values, then press **Fit** again.
5. Use **Plot** any time to look at the spectra.
6. Select parameters and choose **Plot parameters…** to inspect how they vary across the run set (see *Parameters and parameter plots*).

The Quick Start button (`?`) at the right of the tab bar opens a short guide that highlights the section relevant to the active tab. Its sections are expanded into the chapters of this manual.

CHAPTER 2

Getting started

2.1 Prerequisites

muSRFlow wraps external `musrfit` tools. They must be installed and on the `PATH` on the machine that runs the tools:

- Desktop: on the machine running the Electron app.
- Web: on the server that hosts the CGI runner.

The browser edition needs no local installation at all.

2.2 Desktop installation

From a checkout, with Node.js and npm installed:

```
$ npm install
$ npm start
```

npm install is only needed once; later sessions start with *npm start*.

Release packages (Linux RPM and AppImage) are produced with:

```
$ npm run dist:rpm
$ npm run dist:appimage
```

The build and runtime requirements are described in `docs/DESKTOP_DEPLOYMENT.md`.

2.3 Running the desktop app

When the window opens, the **Runs** tab is active and the current year is already selected. Enter run numbers, choose a model, and press **Fit**. The first fit writes the working files next to your MSR document and results appear in the **Results** tab.

2.4 Opening the web edition

Point your browser at the deployment URL (e.g. `http://musruser.psi.ch/muSRFlow/`). Instrument options and run numbers are the ones the server data tree supports. There is nothing to install; the parameter table is also served from the same data tree.

2.5 The Quick Start guide

The `?` button at the right of the tab bar opens a short, tab-aware guide. On the desktop it opens in an independent window; in the browser it opens a modeless dialog next to the main window. Each of its sections corresponds to one tab and is expanded into a chapter of this manual.

2.6 Support

For questions and bug reports contact Zaher Salman mailto:zaher.salman@psi.ch. Issue tracking is linked from the project repository.

CHAPTER 3

Runs and fitting

The **Runs** tab holds everything that selects and describes the data: the run list, the fit range, the fit model, the fit options, and the view settings used for Plot, FFT, and RRF.

3.1 Run selection

Run(s)
: One or more runs, separated by commas. A dash selects a contiguous range and a colon-with-step `from:to:step` produces a regular sequence, for example `520:540:2`.

  Each comma- or range-separated data set is fit independently and merged into one fit document, so `514, 516, 520-525` prepares six data sets.

  A `+` between runs adds the run to the previous data set via the MSR `ADDRUN` directive rather than starting a new data set. `514+515` is a single data set composed of two runs.

  Examples:

  - `514` – one run.
  - `514, 520-525` – two data sets: 514 and each of 520..525.
  - `514+515, 520:540:2` – one ADDRUN data set and a stepped range.

Year
: The data-taking year. The year range starts at 1998 and cannot exceed the current year.

Instrument
: The spectrometer. The exact choice (for example LTF, FLAME, LEM, GPS, GPD, Dolly, VMS, HAL) depends on the data tree your installation exposes.

The path shown in the resolution warning below the row confirms that the chosen runs resolve to real data files; a successful resolution is silent.

3.2 Fit range

- **From / To –** the time window in microseconds that is actually fit (default `0` to `8`).
- **Packing –** the time-binning factor used for the fit (default `100`).

3.3 Fit model

Type

The fit type. **Asymmetry** is the normal choice for a summed forward plus backward count rate. **Single hist** fits a single histogram. The legacy **Asymmetry (Legacy)** variant is kept for old MSR files that used the classic syntax but is not offered for new documents.

Histograms

The detector histograms, or a packed detector name, that make up the asymmetry signal.

An **Asymmetry** fit normally takes one forward,backward pair (for example `21 25`). Several *groups* of one physical run can be fit at the same time by entering slash-separated forward,backward pairs, for example `1,2/3,4` for two groups (detectors 1 and 2 form the first group, 3 and 4 the second). Each group becomes its own RUN block with its own `alpha` and phase, and the groups scale a shared free amplitude by individual ratios (the first group keeps the fixed identity factor). The parameter table carries a per-group `Alpha`, `Phi`, and ratio; the physical model parameters stay shared. Multi-component models and per-run pairs of a multi-run (global) fit are not eligible for the multi-group route.

Components

The theory model is a sum of components. The first component is labelled **Model** and is required; further components are added with + and can be removed with the `x` on their row. Each component has its own parameter set, shown in the **Parameters** tab.

The component list is grouped by physics:

- **Relaxation –** `Asymmetry`, `Exponential`, `Gaussian`, `Stretch`.
- **Static KT –** `SLKT`, `SGKT`, `LGKT`, `STRKT`, `GBGZF`.
- **Oscillating –** `ExponentialCos`, `GaussianCos`, `StretchCos`, `AbragamCos`, `GBGCos`, `ExponentialBessel`, `GaussianBessel`.
- **KT x relaxation –** combined Kubo-Toyabe and relaxation functions, e.g. `LLFExp`, `GLFExp`, `LGKTExp`, `GBGExp`, `STRKTExp` and their spin-flip variants `LLFSExp`, `GLFSExp`, `LGKTSExp`, `GBGSExp`, `STRKTSExp`.
- **Dynamic KT –** `LDKTLF`, `GDKTLF`.

Changing a component rebuilds the parameter table and the constraint preview so their contents follow the new model.

3.4 Fit options

- **Minimizer** – Minuit2 strategy variant used by the fit: `MINIMIZE` (default), `MIGRAD`, or `SIMPLEX`.
- **Errors** – the error calculation run after the minimisation: `HESSE` (default), `MINOS`, `HESSE + MINOS`, or `None`.
- **Strategy** – Minuit2 strategy `0`, `1`, `2`, or **Default** (leave unspecified).
- **Objective** – χ^2 (default) or **Max likelihood**.
- **Save** – write the detailed Minuit2 output including covariance information (on by default).

3.5 Series fit actions

When two or more independent data sets are fit together **without shared parameters**, the fit follows the independent-series route and the **Fit options** row gains a per-run action chip for every data set. This is how a multi-run series is (re)fitted efficiently without re-fitting runs that have not changed.

Clicking a chip cycles through three actions:

- **reuse** – keep the existing fit result for that run.
- **refit** – rebuild the MSR from the live GUI parameters and fit again.
- **skip** – leave the run untouched.

The **All** dropdown applies one action to every chip at once (**Reuse all / Refit all / Skip all**), which is the convenient way to set a uniform policy and then override single runs by clicking their chip.

A series where every run is skipped reports **Nothing to fit**. Sharing any parameter (the **Shared** column in the parameter table) disables this route and the chips disappear; see *Parameters and parameter plots*. The parameter plot is unaffected and still works.

3.5.1 Per-run statistics

For an independent-series fit the status-bar statistic button summarises the series – either **Series did not converge** (some runs failed to converge) or a worst/best aggregate such as χ^2`/NDF worst 1.24 / best 1.01 · 5 runs`. Clicking the button opens a per-run statistics table that lists every data set in run order with:

- the run identifier;
- the action that produced it (**reuse**, **refit**, or **skip**);
- the run's statistic text (χ^2/NDF or max likelihood/NDF, or `Fit did not converge`);
- the converged state;
- the fitted file the value came from.

Reused or skipped runs show the values read from the MSRs already used for plotting. Rows are highlighted for contrast: runs whose value is more than twice the series median stand out as outliers (the spread a worst/best summary might hide), failed runs are marked as not converged, and reused/skipped rows that still carry the on-disk value are visually distinct from freshly fitted ones.

3.6 View settings

Plot, **FFT**, and **RRF** are one-row disclosure buttons; each expands its settings inline, and only one group is open at a time. The dots next to the buttons mark a group whose settings differ from the defaults.

Plot

- **Time range –** independent start and end of the displayed time axis; leaving a field empty selects automatic limits.
- **Y range –** independent displayed Y limits; empty means automatic.
- **View packing –** histogram packaging for the view; empty means automatic.
- **Log X / Log Y –** switch the respective displayed axis to a logarithmic scale.
- **Lifetime correction –** apply the lifetime correction to the displayed spectra (on by default).

FFT

The Fourier-transform display settings:

- **Type –** `Power`, `Real`, `Imaginary`, `Real & imaginary`, or `Phase`.
- **Apodization –** `Strong`, `Medium`, `Weak`, or `None`.
- **Phase –** the phase value or a musrfit phase expression, for example `8.5`, `par7`, `par(7,5,16)`, `parR(...)`.
- **DC correct –** remove the time-domain DC offset before the transform.
- **Range –** the frequency window used for the transform; empty means automatic.
- **Units –** `MHz`, `Gauss` (default), `Tesla`, or `Mc/s`.
- **Power –** the `power` record for the FFT (default `12`).
- **Phase correction –** automatic phase correction `Off`, over the **FFT range**, or over a **Custom range** you type below.

RRF

Rotating-reference-frame display settings. They are strictly opt-in: opening the settings does not change the generated plot block.

- **Enable –** add rotating-reference-frame directives to the PLOT block.
- **Frequency –** the reference frequency, with units `MHz`, `kHz`, `Mc/s`, `Gauss` (default variety shown as Gauss), or `Tesla`.
- **Packing –** the packing for the RRF view (required when enabled).
- **Phase –** optional phase in degrees, or a parameter reference such as `par4`.

3.7 Fit and Plot

The two toolbar actions work on the current window state:

- **Fit –** validate the run selection, prepare the MSR document from the GUI, and run `musrfit`. The fitted values flow back into the parameter table and the results.

- **Plot** – validate the run selection, prepare the same MSR document, and run `musrview` to display the spectra. On the web the resulting ROOT canvas is drawn inside the page window with JSROOT; on the desktop `musrview` opens its own native window.

While either action runs, the status bar shows progress and the Results tab accumulates the command log. Plot-compatible values are cached so a plot does not lose the numerical fit quality already returned by a previous fit.

On the desktop, a local **Fit** uses every available CPU core by default. Set **musrfit threads** in *Preferences* to cap the OpenMP thread count (`musrfit -u`) when a machine is shared or a fit would otherwise saturate the login node; leaving it blank keeps the all-cores behaviour. The web edition always fits on the server and ignores the setting.

If the MSR editor owns the document (a hand-edited MSR is authoritative), the editor block is used for Fit and Plot as-is and GUI changes do not silently replace it – see *MSR editor*.

CHAPTER 4

Parameters and parameter plots

The **Parameters** tab shows the numerical model: one row per fitted parameter, per data set, with starting values, steps, and limits. Fitting feeds the measured values back, and the same table drives the interactive parameter plots.

4.1 The parameter table

The table is rebuilt automatically whenever the model changes (a new component, a new run, a constraint). Rows are keyed to the model:

- Single-run fit: one row per parameter of the selected model.
- Multi-run fit: a row per parameter per data set, each named by its run (for example `Lam_2255`), plus the parameters you choose to share.

Rows follow the physical role of each parameter: the initial asymmetry first, then the relaxation/lineshape parameters, then the precession frequency, then the phase, then the asymmetry alphas and group ratios, and finally the detector normalization/background parameters. This ordering is what ends up in the generated `FITPARAMETER` block and therefore in the `*_par.dat` columns from `msr2data`. A parameter introduced through a constraint (for example `Bfield = gamma_mu * Frq`) is placed with the built-in parameter it is linked to; when it is linked to several parameters it takes the highest-priority (lowest) rank among them. An order imported from an existing MSR file always overrides this physical sequence.

For each parameter the columns are the **starting value**, the **step**, and the **lower/upper limits**. Edit any of them directly. Run-scoped parameters are matched to their data set by the run suffix, so a per-run starting value only affects that run. The values in this table are the starting values used for the next fit; after a fit they are replaced by the fitted results, and the fit uncertainties are available in the parameter plots as error bars.

4.1.1 The Shared column

In a multi-run fit the table also has a **Shared** column in the first row of each parameter. Tick it to keep a single shared starting value, step and limits for all runs instead of a separate per-run copy. This is equivalent to the musrfit `GLOBAL` parameter mechanism.

Sharing any parameter takes the fit out of the independent-series route, so the per-run refit/skip action chips disappear (see *Runs and fitting*). The parameter plot still works.

4.2 Selecting parameters for plotting

Click a parameter name to select or deselect it for plotting:

- **Click** – toggle that single parameter.
- **Ctrl / Cmd-click** – add an individual parameter without changing the rest of the selection.
- **Shift-click** – select a contiguous range of parameters.

The selection state is shown in the row. Then use the **Plot parameters…** button at the top of the parameter table to open the plot window. If the parameter table is not yet populated (no fit run yet and no file loaded), the plot can still build the parameter data inline from the table – the button stays enabled whenever the table is built.

4.3 The Choose X axis dialog

The dialog lists the parameter data columns and asks which one to use as the common X axis. Each selected parameter becomes a Y series.

The dialog also offers the **Layout** of the plot window:

- **Auto** – one parameter per panel. Up to three parameters share a single column; beyond that the panels grow toward a square grid (two parameters in two columns, three in three, and four or more arranged as the closest square grid).
- **Single row** – all parameters side by side in one row.
- **Single column** – all parameters stacked in one column.

The layout choice is remembered for the next plot. The window opens with the chosen arrangement, and the **Plot options** panel lets you re-arrange the panels afterwards.

4.4 The parameter-plot window

The window shows one panel per data set / parameter as arranged by the layout choice. Panels share the chosen X axis. The toolbar holds:

- **Refresh** – reread the current parameter-data file and redraw every visible panel. Use this after a new fit to update the numbers.
- **Data…** – change the data set loaded in the *active* panel.

- **Add data set… –** choose one or more parameter-data files and overlay them onto *every* panel at once. Each file keeps one colour from the Palette, and every panel plots the parameters it already shows for each newly added file, so a scan can be compared against the current one across the whole window with a distinct colour. The Y-axis label stays on the active panel's parameter rather than reverting to a generic label. Adding files this way is the same overlay mechanism as **Overlay data sets** in *Plot options*; the toolbar shortcut just applies the additions to all panels instead of the active one.

Overlaid curves get a *smart legend* label that says which physical state each scan represents instead of only a file name. The label is derived from the data set's own scan variable and its constant coordinate: an energy scan (`dataE` varying) is tagged with its fixed temperature (`(T = 300 K)`), a temperature scan with its fixed energy for LE-µSR (`(E = 12 keV)`) or with its field for bulk µSR (`(B = 100 G)`), and a field scan with its fixed temperature. When no fixed coordinate is present the plain file label is used. You can also rename a legend entry directly in the plot; a Refresh then keeps your name instead of rebuilding the default.

- **Plot options –** expand the layout, X, parameter, style, and displayed series controls for the active panel.

4.4.1 Plot options

- **Layout –** the concrete panel grid (for example `1 x 1` … `3 x 3` and larger grids as the panel count needs). Changing it adds or hides panels; new panels start empty and clone the sync settings of the current panel.
- **Panel indicator –** which panel is active (`Panel 1/3`). Click a panel to activate it before changing its panel-specific settings.
- **Data set –** the data source shown in the active panel.
- **X –** the X column for the active panel.
- **Parameter –** add a Y parameter to the active panel. Serial chips underneath list the current Y series; click the `x` on a chip to remove that series.
- **Style –** `Scatter`, `Lines + markers`, or `Lines`.
- **Palette –** the base colour palette (`Default`, `Bold`, `Pastel`, `Monochrome`) used to assign each overlaid data set its default colour. Changing it restyles every panel.
- **X title / Y title –** override the automatic axis label of the active panel; leave empty to keep the automatic label.

4.4.2 Remembered plot configurations

The full window state is remembered per data source: the panel arrangement, the X column, the template, the palette, the per-series styles (including the error-bar caps setting), and the overlay files. A configuration is keyed by the data source *and* the parameter set it plots, so each parameter selection has its own remembered configuration.

Restoring a configuration is always an explicit choice, never something that happens behind your back:

- The **Choose X axis** dialog appears every time you plot parameters.

- If a remembered configuration matches the parameters you selected (the same set, in any order), a **Configuration** dropdown appears in the dialog. The most recent matching configuration is preselected; the dialog shows exactly what will be restored, and you press **Plot** to confirm.
- Choosing **From scratch** in the dropdown (or having no matching configuration) starts a normal new window from the X axis and layout choices on the dialog.

Matching is exact: a saved configuration for three parameters never shows up for a one-parameter request, even if that one parameter is part of the three.

- State is keyed by the data source, so `scanA_par.dat` keeps its own styles and layout while other files (and their overlay sets) remember theirs independently.
- Up to eight data sources are remembered; adding a new one forgets the oldest. Each source keeps at most one configuration per parameter set (the newest wins) and a handful of parameter sets in total.
- **Forget plot configuration** in **Plot options** clears the remembered state for the data source shown in the active panel, so the **Choose X axis** dialog offers no configuration next time.
- The remembered configuration of the current data source is carried in configuration share links, so a colleague restoring your link gets the same plot window – see *Configuration sharing*. When a real fit produced the parameter table, the table itself is embedded in the link (inline text) and a web restoration renders the identical plot immediately, without needing a local or server-side fit.
- Uploaded files keep only their identity, not their contents: if the original file cannot be read again (for example it was deleted), the restored plot degrades gracefully and the dialog is offered instead.

4.4.3 Overlay data sets (multiple files in one panel)

Panels can plot several `*.dat` files at once so scans can be compared on one set of axes. Open **Plot options** and use **Overlay data sets > Add file…** to choose extra parameter-data files (on the desktop this is a native multi-select dialog; on the web it is the browser's multi-file picker). Each file becomes a row with a checkbox:

- checked rows are drawn on top of the panel's main data set;
- the colour swatch shows the default trace colour for that file, assigned from the **Palette**;
- files are identified by their resolved path or blob identity, so picking the same file twice keeps one entry – and one user-adjusted style;
- the X axis is aligned across files **by column name** – a file that does not contain the panel's X column is skipped and named in a striped note under the panel;
- in overlay mode each file's traces share its colour, and legend entries are prefixed with the file name (`scanB_par.dat: λ1`).

Growing the layout clones the active panel, so expanding a `1 x 1` grid to `2 x 1` carries the overlay files into the new panel. Unchecking a row removes that file from the active panel only; the file stays available in the list for other panels.

4.4.4 Editing trace styles

Click any marker directly in a plot panel to open a small style editor near the pointer:

- **This trace / This data set** – whether the change applies to the clicked (file, parameter) series only, or to every trace of that data set (all panels showing it). Choosing **This data set** unifies the whole file and clears any per-panel trace styling left behind, so those panels no longer shadow the new style.
- colour swatches, marker **symbol**, **size**, and **opacity**;
- **Error-bar caps** – error bars are plain vertical lines unless the checkbox is ticked, which draws the cross-bars at both ends;
- **Apply** – commit the staged choices and redraw the affected panels; **Close** hides the editor, and ESC (or `Q`) dismisses it;
- **Reset** – restore the file's default colouring for that scope.

Trace-level edits are keyed by (data set, panel, parameter), so they survive panel re-layouts but never leak onto other panels. Data-set-level edits are shared: recolour `scanA_par.dat` once, and every panel that shows it updates. `Esc` (or `Q`) closes the editor; the title bar is a drag handle so the panel can be repositioned within the parameter-plot window.

4.4.5 Template and series presets

The **Template** dropdown applies a predefined multi-panel arrangement that maps recognised parameter names to roles. For example the `tf-scan` template lays out three panels – amplitude, damping, and frequency – above one another. Parameters that cannot be matched to the data are omitted rather than guessed; unmatched panels stay empty.

4.4.6 Keyboard shortcuts

With a panel focused (and not while typing in a field), the shortcut keys are:

- `u` – reset the active panel to its automatic X and Y ranges.
- `Q` or `Esc` – close the plot window.

The plotly mode bar at the top-right of each panel provides zoom, pan, box select, and download, and a custom button toggles each axis between linear and logarithmic. Trace colours are assigned automatically per series – clicking a marker opens the style editor described above to change them.

CHAPTER 5

Constraints

The **Constraints** tab adds fit functions and parameter relations that are not represented by the main theory controls. It is split into three parts:

- **Custom parameters –** the named constants you define once and reference from constraints.
- **Constraints –** one row per constraint: a parameter, a function expression, and an Add button.
- **Preview –** a live read-only rendering of the `THEORY` and `FUNCTIONS` blocks that the MSR will contain.

Everything on this tab is included the next time the MSR is generated, so you can refine the preview without running musrfit.

5.1 Custom parameters

Type a name (for example `Bfield`) and press **Add**. Custom parameters become available on the parameters list and can be referenced by constraints like any ordinary fit parameter. They are removed by the same workflow that removes a normal model parameter.

5.2 Constraints

A constraint binds the value of one parameter to an expression of others. To add one:

1. Choose the parameter to constrain from the **Parameter** dropdown.
2. Type the expression into the **Function** field.
3. Press **Add**.

The expression uses musrfit `FUNCTIONS` syntax: arithmetic on parameter names and literals.

5.3 Example: couple the magnetic field to the Larmor frequency

A typical use is to avoid fitting a frequency directly and instead fit an applied field:

1. Add a custom parameter named `Bfield` under **Custom parameters**.
2. Under **Constraints**, choose `Frq` from the **Parameter** dropdown.
3. Type `gamma_mu * Bfield` into the **Function** field and press **Add**.

The constraint reads "the Larmor frequency `Frq` equals the muon gyromagnetic ratio multiplied by `Bfield`". The theory and functions preview updates immediately so you can verify the generated `FUNCTIONS` text before fitting.

5.4 Preview

The **Theory** and **Functions** panes show exactly what the MSR will contain. The Theory pane reflects the model components from the Runs tab; the Functions pane reflects the constraints. Both are read-only – use the GUI controls to change them.

CHAPTER 6

t0 / background

The **t0 / bg** tab hosts expert controls used when detector timing or background behaviour must differ from the normal defaults.

Leave these controls unchanged for standard fits. When you do adjust them, they are written into the generated MSR like any other setting and take effect on the next Fit or Plot.

The tab loads its controls from the current MSR document, so editing an existing document and returning to this tab keeps showing the values that document actually uses.

6.1 Multi-group asymmetry fits

For a multi-group asymmetry fit (`1,2/3,4` in the **Histograms** field, see *Runs and fitting*) the groups share the single row of their one physical run, and a field holds one value per group in group order. When importing such an MSR, values that agree across the groups collapse into the tab's common fields and distinct values stay per group (for example `t0 forward 101 103` for two groups). A file whose groups disagree in an inconsistent way - some groups with a `t0` and others without, mixed background modes (a group cannot have both a background range and a fixed background), or a good-bin data range present for only some groups - is rejected without modifying the previously stored values, since a guessing read-back would shift every later group's token.

CHAPTER 7

MSR editor

The **MSR Editor** tab displays the complete MSR file that muSRFlow prepares. It is a real editor, not just a preview: you can edit the text by hand, and the edited document becomes the source of truth for the next Fit or Plot.

7.1 Ownership: GUI or editor

Two actors can own the document:

- the **GUI**, which regenerates the MSR from the current tab settings; and
- the **editor**, once you edit the text by hand.

The lock icon in the editor toolbar shows which one owns the document. An open padlock means the GUI owns it and the next GUI generation or fit can replace the text; a closed padlock means the editor owns it and Fit and Plot use the text as-is.

7.2 Workflow

GUI to MSR (the `GUI -> MSR` button)
: Replaces the visible document with one generated fresh from the analysis settings. The GUI is the source of the text.

Edit by hand
: Entering text hands ownership to the editor: the lock icon becomes a closed padlock. While locked, **Fit** and **Plot** use the document as-is and GUI changes do not silently replace it. Any attempt that would overwrite the edited text asks for confirmation first.

MSR to GUI (the `MSR -> GUI` button)
: Copies the recognised settings of the current document back into the GUI fields (run list, fit range,

model, parameters, and constraints). This does not surrender editor ownership; the editor keeps the document until you unlock it.

Unlock (click the closed padlock)
Returns control of the document to the GUI, so the next GUI generation or fit may replace it.

7.3 File handling

On the desktop (Electron), the **File** menu opens and saves MSR files directly. The **Save** button in the editor toolbar writes the current document to disk. On the web, saving is handled through the server's result artifacts.

The editor also shows which source document it currently displays. For a multiple-run fit it can expose the generated document and the template it was built from; the active tab (**MSR** or **Template**) selects which one is shown.

CHAPTER 8

Results and diagnostics

The **Results** tab collects what each Fit or Plot produced: the generated files, the execution details, and the full command log. The status bar at the bottom of the main window gives the short-lived state without opening the tab.

8.1 Result files

The primary artifact is the **Result MSR** – the fitted document that contains the converged parameter block. Under **Other files** the tab lists:

- **ROOT plot** – the ROOT canvas produced for the spectra display.
- **Template MSR** – the template the document was generated from.
- **Output log** – the full output of the run.
- **Parameter data** – the `par.dat` parameter table that feeds the parameter plots.
- **All files** – a link to the whole artifact package.

On the desktop the links open the files on your machine; on the web they download or display the server-side artifact.

8.2 Execution details

A collapsible **Execution details** block records when the action started and finished, the process exit code, and whether the GUI parameter table and the fitted result were synchronised afterwards.

8.3 Command log

The command log is the authoritative record: every command line muSRFlow invoked, in order, with its arguments. It appears in the tab and in the pop-up run log explained below. Use it to reproduce an analysis by hand or to diagnose unexpected behaviour.

8.4 The status bar

The status bar under the main window keeps a terse permanent status:

- the **state** button (for example `Idle`, `Running musrfit`) – click it to open the run log immediately;
- the **fit statistic** (for example χ^2`/NDF 1.012`), kept from the fit so a later plot does not erase it. For an independent-series fit the button instead shows the worst/best aggregate of the per-run statistics (or `Series did not converge` when any run failed); clicking it opens the per-run statistics table – see *Runs and fitting*;
- the **message** line for transient notices (click it to open the run log);
- the **file** and **directory** the last action operated on.

Hovering over a shortened value shows the full text. The state, statistic, message, file, and directory elements appear only when they add context.

8.5 Run log window

The run log dialog shows the same command log as the Results tab, as a large readable pane. It is opened from the status bar state or message, from the outcome dialog, or by clicking the log area in Results. It is modeless, so the main window stays usable while it is open.

8.6 Outcome dialog

If an action cannot be completed (a failed fit, an invalid run selection), an outcome dialog explains what went wrong. It offers **Close** and, when relevant, **Open run log** to inspect the commands that ran. A successful action does not pop this dialog – it stays out of the way.

CHAPTER 9

Configuration sharing

muSRFlow can encode the current analysis settings into a single link that another instance can restore. This is useful for reproducing a session or for passing a fit setup to a colleague.

9.1 What is shared

The link carries the run selection, instrument, year, fit range, model components, fit options, parameters and constraints, plot/FFT/RRF view settings, the analysis-related preferences, and – when a parameter plot configuration was remembered for the current data source – the parameter-plot window state (panel arrangement, palette, trace styles, overlay files). When the sender's parameter data is available on the machine that produces the link (a completed fit), the actual msr2data table is embedded as well, so the recipient can draw the identical plot – fitted values, scan variable, and error bars – without reproducing the sender's `*_par.dat` on its own file system. It does **not** carry the raw data files or your local working directories. A restored session therefore recreates the analysis on top of whatever data the target machine can see.

9.2 On the web

The share button in the tab bar (the chain-link icon) copies a link such as

```
http://musruser.psi.ch/muSRFlow/#c=...
```

Anyone who opens the link in a browser gets the same analysis settings, as long as the server exposes the same data.

A visitor who has the desktop build installed can also use the second button next to the share button (the monitor icon). It copies the same payload as a `musrflow://` link instead:

```
musrflow://open/#c=...
```

A browser cannot navigate to its own `https` page and then hand the payload to an installed application, so this dedicated scheme lets the web page open the settings directly in the desktop app. Only the base URL differs; the encoded fragment is byte-for-byte the same, so either link restores the same analysis.

9.3 On the desktop

Electron cannot handle its own browser links, so it ships the same encoding backwards as well: open a muSRFlow configuration link (from the web or from a colleague) and choose **Restore configuration link** to paste it into the restore dialog. The analysis settings are applied while your local data and working directories remain untouched.

The installed application also registers the `musrflow://` scheme, so clicking such a link (in a mail client, a document, or the web page above) starts muSRFlow and restores the embedded settings directly. The canonical `http://musruser.psi.ch/muSRFlow/#c=...` link still works: paste it into the restore dialog. See the desktop deployment notes for how the scheme is installed, including the AppImage caveat.

9.4 Notes

- The shared payload is a compact, optionally compressed fragment in the URL hash; nothing is sent to a third party.
- A restored configuration may reference data that was never fit on the target machine. The parameter table is rebuilt from the restored settings and can be populated by running a fit.
- The parameter-plot stanza is optional and keyed by the data source's identity. When the sender's parameter table could be read at share time it is carried inline as the main source, so on the web the plot opens immediately after restore (the browser's `Plot parameters...` action applies it the first time it is used). Otherwise it is applied when the same source exists on the target machine – a matching on-disk parameter-data path or an identical inline parameter table (see *Remembered plot configurations*). If neither applies it is ignored harmlessly and the plot opens with the normal dialog.

CHAPTER 10

Preferences

The **Preferences** dialog is a desktop (Electron) feature: it opens from the **File** menu. The web edition derives the same settings from server configuration and never exposes this dialog.

Work directory
: Where generated files, MSR documents, and result artifacts are written. Defaults to the current directory (`.`).

Data directory
: The root of the instrument data tree (for example `/psi.ch/group/lmu/public/data`). Run paths are resolved under it.

musrfit startup XML
: The `musrfit_startup.xml` used to resolve portable run names. Leave it as **auto-detect** and muSRFlow looks for the standard installation file; choose a specific file when you run a specialised setup. A hint under the field shows the naming mode the detected XML enables.

musrfit binaries
: The directory that holds the musrfit tools (`musrfit`, `musrview`, `msr2data`, and `dump_header`). Leave it empty for **auto-detect** and muSRFlow searches the usual installation locations; choose a directory when the tools live somewhere non-standard (for example a developer build tree). The configured directory wins over every auto-detected location for the tool commands muSRFlow runs, and is put first on the `PATH` given to them. The setting is a desktop feature: it is remembered across sessions, while the web edition takes its tools from the server and never exposes this field.

musrfit threads
: The number of OpenMP threads passed to a local `musrfit` run (`-u`). Leave it blank to let musrfit use every available core. Set a lower number to keep a fit from saturating a shared workstation or login node. The value is validated by musrfit (`1 ... number of cores`) and has no effect on the web edition, which runs fits on the server.

Initial parameters
: **Guess initial parameters from first-run header** (on by default): when a run set is added, muS-

RFlow reads the first run's metadata to seed sensible starting values (for example an approximate frequency or asymmetry) instead of generic defaults.

Output file suffixes
: The suffixes appended to generated artifacts: **Asymmetry** (default `_asy`) and **Single histogram** (default `_singleHist`). Change them to avoid clashing with existing files or to match a naming convention.

Save stores the values for the next session; **Cancel** closes the dialog without changing anything. Directory changes take effect from the next action that resolves or writes files.

CHAPTER 11

Web and desktop editions

Both editions share the tabs, fit engine, model library, parameter plots, and configuration-sharing protocol. The differences come from where the data and the command-line tools live.

Concern	Desktop (Electron)	Web (browser)
Where the tools run	Locally via the `PATH`.	On the server behind the CGI endpoint.
Data access	Any local directory under the configured data root.	What the server data tree exposes.
Preferences	Per-user, through **File / Preferences**.	Fixed by server configuration; no dialog.
Opening / saving MSR files	Native **File** menu and editor Save button.	Server-side artifacts; no local file dialog.
Spectra display	`musrview` in its own native window.	ROOT canvas produced by `musrview` on the server, drawn with JSROOT in the page.
Quick Start help	Independent auxiliary window.	Modeless dialog beside the main window.
Configuration links	Share webpage links; **Restore configuration link** to paste one.	Share button copies a link; opening it restores the settings.

The workflow chapters of this manual apply to both editions. Where a step is specific to one edition (handling files, Preferences), the text says so explicitly.

CHAPTER 12

Keyboard shortcuts

Most of muSRFlow is mouse-driven, but a few shortcuts speed up the everyday tasks.

12.1 Parameter selection (Parameters tab)

- **Click a parameter name** – select or deselect it for plotting.
- **Ctrl / Cmd-click** – add just that parameter to the selection.
- **Shift-click** – select every parameter between the anchor and the one clicked (holding Ctrl as well adds the range).

The same actions work from the keyboard: focus a parameter name and press **Enter** or **Space** to toggle it.

12.2 Parameter-plot window

With a panel focused (and not while typing in a field):

- `u` – reset the active panel's X and Y axes to automatic ranges.
- `Q` or `Esc` – close the plot window.

12.3 MSR editor

The editor is a standard editable document: the usual cursor, selection, and clipboard keys apply. Its ownership state depends on whether the text has been edited by hand; see *MSR editor*.

12.4 General notes

- Shortcut keys ignore events typed into input fields, selects, and text areas so they never collide with normal text entry.
- In the parameter-plot window, plotly's mode bar provides the full set of interactive chart tools (zoom, pan, box and lasso select, downloads).

CHAPTER 13

Troubleshooting

13.1 Fit or plot will not start

Check the resolution warning under the run row: it reports runs that cannot be resolved against the current data tree. In the browser edition, confirm the server exposes those runs; on the desktop, verify the data directory preference points at the real data root. If a model has no valid run, the action reports the reason in the outcome dialog and the run log.

13.2 “Nothing to fit”

An independent-series fit (two or more data sets, no shared parameters) where every run is set to **skip** reports this instead of running. Set at least one chip to **reuse** or **refit**, or use **Refit all** from the All dropdown.

13.3 Fitted parameters look wrong or unchanged

- The parameter table is the source of the *starting* values. After a fit it holds the fitted results; run the fit again after editing values.
- Hand-edited MSR documents are authoritative: if the editor lock is closed, Fit and Plot use the editor text and ignore GUI changes until you unlock (see *MSR editor*).
- A model change rebuilds the parameter table and may reset values; re-check the rows for every component you use.

13.4 Parameter plot is empty or missing data

- Click **Refresh** after a new fit – panels reread the parameter file.
- Use **Data…** to change the data set of the active panel.
- A panel that shares the X axis with no plottable column shows its empty message; activate it and add a parameter or load another file.
- Overlaid files that lack the panel's X column (or could not be re-read) are skipped and listed in a striped note under the panel. Fix the X column (**X** control) or uncheck the file under **Plot options > Overlay data sets**.

13.5 Parameter plot hangs on "Preparing plot files"

"Preparing plot files" covers the setup performed before the tools run. If the step never finishes, check the server/work-bench side:

- The command-line tools must be reachable on the machine that runs them (local install on the desktop, server-side install for the CGI).
- On the desktop, confirm the work directory is writable.

The command log records every command line that was attempted; it is the first place to look.

13.6 Web page cannot load data

The browser edition fetches data and runs fits on the server. If the page loads but actions fail, verify the CGI endpoint is reachable (see `cgi/README.md`) and that the server data root matches what the page was deployed with. The server health endpoint reports the configured data root; muSRFlow's resolution warning should agree with it.

13.7 Restored configuration differs from the original machine

Configuration links carry analysis settings only, not data or working directories. On a different machine the same settings resolve against that machine's data tree. If runs do not resolve, fix the data/startup-XML configuration (desktop Preferences; server configuration on the web) and refit.

13.8 Some links start with a different prefix

muSRFlow shares both a compact compressed form (`#c=`) and a readable legacy form (`#cfg=`). Both are accepted by the web and the desktop restore dialog.

13.9 Reporting problems

Include the command log (open from the status bar or the outcome dialog) and the versions involved. Contact Zaher Salman mailto:zaher.salman@psi.ch with those details for the fastest help.

CHAPTER 14

Glossary

MSR
: The `musrfit` input document. A text file made of blocks – `FITPARAMETER`, `THEORY`, `FUNCTIONS`, `GLOBAL`, `RUN`, `COMMANDS`, `PLOT`, `FOURIER`, `STATISTIC`. muSRFlow generates, edits, and fits this file.

fit document
: The prepared MSR plus the parameters, constraints, and settings that describe one fit. The MSR Editor shows it as text.

RUN
: A data set record inside an MSR which names one measured run or a single `ADDRUN`-joined set of runs.

ADDRUN
: An MSR directive that appends another run to the current data set, so `514+515` forms one data set from two runs.

independent-series fit
: A multi-run fit in which every data set has its own parameters (no `Shared` entries). muSRFlow treats it specially: it can reuse, refit, or skip each run independently through the action chips.

global parameter
: A parameter marked **Shared** in the parameter table. One starting value, step, and limits apply to all runs, matching musrfit's `GLOBAL` mechanism.

par.dat
: The parameter table produced from a fitted MSR (by `msr2data`). Each line is one run and each column one parameter (plus its uncertainty); the parameter-plot window reads these files.

musrfit
: The muon-spin-spectroscopy fitting engine. See the musrfit project pages for its manual.

musrview
: The reference viewer for muon-spin data and fits. muSRFlow runs it for the spectra display and for ROOT canvases of fitted data.

msr2data
: A companion tool that writes the `par.dat` parameter table from a fitted MSR.

dump_header
: A companion tool that prints run metadata. muSRFlow uses it to resolve portable run names and to seed initial parameter guesses.

ROOT plot
: A plot canvas produced by `musrview --root`. On the web it is drawn with JSROOT in the browser; on the desktop it is part of the native viewer's output.

RRF
: Rotating reference frame. An optional display transform that subtracts a reference frequency from the measured signal; configured in the RRF view settings.

FFT
: Fast Fourier transform. The frequency-domain view of the signal, configured in the FFT view settings (type, apodization, phase, range, units, and automatic phase correction).

configuration link
: A URL carrying the current analysis settings as a compressed fragment. Both the web share button and the desktop restore dialog use the same format.

If anything in this manual does not match the behaviour you see, contact Zaher Salman.

Separate repository documents cover desk build and packaging (`docs/DESKTOP_DEPLOYMENT.md`) and the web-server/CGI deployment (`cgi/README.md`); this manual concentrates on using the application.

Index

A

ADDRUN, 39

C

configuration link, 40

D

dump_header, 40

F

FFT, 40
fit document, 39

G

global parameter, 39

I

independent-series fit, 39

M

MSR, 39
msr2data, 40
musrfit, 39
musrview, 40

P

par.dat, 39

R

ROOT plot, 40
RRF, 40
RUN, 39