

Womo-Reise — User Manual

What is Womo-Reise

A database (SQLite) with forms to interact with that database. It is vehicle-specific, meaning each vehicle has its own database that can be switched or archived at any time.

The project has grown quite large, so a few bugs may still show up while working with Womo-Reise. Please report them to me so I can fix them.

What can you do with Womo-Reise

Roughly speaking, manage all the data that comes up during a trip with the motorhome. Specifically:

- Manage several vehicles in parallel
- Keep a travel diary (Daily leg)
- Keep a fueling log — fuel, LPG, and AdBlue
- Archive travel expenses
- Create consumption and cost calculations
- Manage contacts (for example companies, repair shops)
- Embed external files such as photos, PDFs, and navigation files
- Back up and restore the database (Backup / Restore)
- Selectively download a trip's data to share it with friends

User manual

First start

When you launch Womo-Reise for the first time, the form for creating a vehicle opens automatically. This is not by accident: Womo-Reise is vehicle-based — each vehicle has its own database. No other module works without an active vehicle, so this is the first and only required step after the installation.

Enter the vehicle's master data (brand, model, license plate, technical data). The most important entries here are the fuel types — diesel or petrol, optionally also gas (LPG) and AdBlue. Your choice controls what you will see later in the fuel module. For example, if your motorhome has no AdBlue tank, this option will not even appear in the fuel form.

Once the vehicle is created, the main screen appears with all the modules. From

now on you can work freely.

If you want, you can create a first trip right away — via Main screen → Travel diary → create / delete new trip. But this is not required. Fuel entries and expenses can also be recorded without a specific trip; they are then assigned automatically to the default category “General — no trip”. This default is always present and cannot be deleted. It collects all data that does not belong to a particular trip, for example fuel entries during everyday driving between trips.

If you want to know in more detail

A look under the hood at how a database works: the heart of every database is its individual tables. That's where all the entered data is stored. You could enter the data into the tables by hand, but that's very inconvenient. So forms are developed for comfortable entry and querying of the data. The form itself stores no data, it is only the mask, the template through which you enter the data conveniently. The data itself is stored in the tables of the database. In this case, it's a SQLite database.

It is what's called a relational database, which simply means that the individual tables are linked to each other (relations). This is what makes complex queries possible in the first place — for example, calculating fuel consumption when refueling. Data from three different tables is queried and linked together.

Sounds complicated, but you don't need to worry about any of this when working with Womo-Reise.

Now to Womo-Reise itself

When you start the program, the start page appears. From there, you reach the individual forms.

First, some general information about the various windows and buttons.

- Some fields (often dates) require an entry, because saving without them wouldn't make sense. Usually there's a corresponding hint label. The program also speaks up when you click Submit data and have forgotten a required field (an error message appears).
- Other fields don't accept input — they only display values. These usually have a gray background.
- Yet others are selection menus, meaning you can only choose from preset values. The menus draw on tables that are pre-filled with values (for *example cost category, payment method, travel expenses*). The restriction to pure selection is meant to protect these tables from accidental changes. For the presets I picked the values that seemed most useful to me. Of course these tables can be changed or extended too, but only via: *Start menu* → *Edit selection menus*.

- Then there are combo boxes. They allow both entry and selection of existing data, for example for *Locations* in the form *Daily leg*.
- There are also various navigation buttons. They are usually colored differently (for example *Submit data*, *Back to start page*, *Calculate daily km*).
- Often, there are tables at the bottom of a form summarizing the most important data. Double-clicking a field of a record loads it into the input fields for editing or deletion. Editing or deletion isn't possible for all fields, because otherwise the database would become inconsistent and the queries would no longer be correct.

Attention: When deleting, there is a confirmation dialog ("Are you sure..."), but once confirmed the record is gone for good.

- A special case are the sections or forms where you can embed external files, for example *PDF documents*, *photos*, *navigation files*. The files themselves are not stored in the database, only a link to them. This keeps the database nice and lean. For this to work, you need to stick to the predefined paths.
- Within the program folder there is a directory called *PDF* which is further divided into *Photos*, *Notes*, and *Navigation*. The actual files are stored there. The forms then hold the paths to those files. When you click an upload button, a dialog opens showing your computer's directory structure for picking files.

The **start page** is divided into several sections. We'll go through some of them.

Vehicle management

Womo-Reise is vehicle-specific — each vehicle has its own database. When you start the program for the first time, you have to create a vehicle first, otherwise you cannot use the app.

A selection in the fields *Designation*, *Main fuel*, *AdBlue*, and *LPG* is required. The created vehicle appears under *Active vehicle* after restarting the app.

Later you can certainly manage several vehicles in parallel — for example keeping the old motorhome as an archive while using the new one as the active vehicle.

You reach the Vehicle management form via Start page → Other → Vehicle management. Here you can:

- Create a new vehicle (for example after buying a new motorhome)
- Switch between vehicles, if multiple are set up

- Archive a vehicle — the data remains as a read-only archive but is no longer active
- Maintain a vehicle's master data (make, model, license plate, technical data)

Vehicle switch

Important to understand: when switching between vehicles, it's not just a selection that changes — the active database itself is switched. From that point on, all forms display the data of the newly selected vehicle. During the switch, the previous active vehicle's data is saved as a snapshot (see Archive below), so nothing can be lost.

Archives

When you have given away or sold a vehicle, you can archive its data. The archive is a read-only copy of the database at a specific point in time — you can look into it but no longer change anything. This protects the historical database from accidental edits.

Attention: Deleting an archived vehicle removes the archive permanently. Please think it through carefully whether you really no longer need the data.

Contacts / addresses

The contacts you collect on the road are stored centrally in a table. You open the form via the *Contacts* button.

You can recall already-stored contacts via the selection list and edit or extend them if needed (for example, add a memory photo). Contacts are organized by *trips* — that's why selecting a trip is mandatory, as is entering a last name. All other entries are optional.

All stored contacts are shown in the table *Stored contact data*. Double-clicking an entry loads the data into the input fields for editing. This also works by selecting an existing name in *Contacts – selection list*.

Trip — the centerpiece

Daily leg

Daily leg is a kind of road book: here you can enter the basic data for each travel day, such as date, kilometers, and much more. Daily and total kilometers are calculated when you click. The total km are calculated specifically for each trip, so there can be no mix-ups. To create a new trip, use the **form Create new trip** in the start menu.

Required entries are: *Trip selection, date, start and end kilometers, overnight*

location. Clicking the buttons Click for *daily kilometers* and Click for new *total kilometers* is also required. With this, the basic data for a travel diary is saved.

All other entries are again optional.

Safeguards against incorrect entries

If one of the required input fields is not filled in, or if the kilometer buttons have not been clicked, the program issues error messages and saving is not possible.

If the date is not greater than the previously entered date, a warning appears. Likewise for incorrect *kilometer entries* (end kilometers smaller than start kilometers, or unusually much larger, more than 2000 km).

Don't be surprised that you can't enter just anything — this also serves safety. For example, *kilometer entries* only accept whole numbers, no letters. Other fields have similar safeguards (*costs*, *navigation entries* etc.).

The form integrates several extra modules: *Weather*, *Navigation*, *Photo*, *Notes*.

Weather

Here you can enter some basic weather data, including a more detailed description in *Weather detailed*. For *Short and sweet*, single or multi-selection is possible.

Navigation

If your computer is connected to the internet or has GPS reception, clicking *For location determination* automatically determines the location and enters it. Otherwise, the data has to be entered by hand (for example from the GPS device) in the specified format. If the data is in a different format, it can be converted by clicking *Convert coordinates*. Operation is essentially self-explanatory.

By clicking *Select nav file*, recorded navigation data can be embedded. This is mainly intended for waypoints or tracks that you have previously recorded and downloaded from the GPS via a navigation program. Almost every nav program (for example QuoVadis) has an export function. Use that and pick the format .gpx or .kmz (Google Earth). Almost all programs understand these, and they are by far the most common. The data is stored in the integrated program folder *PDF/Navigation*, and only the link to it ends up in the database.

Photos

Maybe you'd like to add a photo of the overnight location for better memory. This is possible too. You can pick photos (multiple if you want) from the computer's file system, and the link to them is saved in the database. The actual photos go into the integrated program folder.

Photos should ideally be downscaled, that is, made smaller, before saving, in order to save storage space.

The selected photos are shown before saving. All previously saved photos can be viewed in the *photo gallery (show stored photos)* — *trip, location, and date* are noted there.

Notes

Maybe in addition to or for every day you'd like to keep a detailed travel diary (for example in Word, LibreOffice, plain text). These too can be picked from the computer and saved alongside, similar to photos. In principle many formats can be saved, but it's recommended to convert texts to *PDF* before saving and embed them as such (almost all word processors do this nowadays; otherwise there are free PDF converters).

Common to all extra modules (except Weather)

- You can pick multiple files from the computer's file system at once.
- Before saving, the file is shown (except for Navigation).
- The files can be edited. More on that below.
- The files themselves can be copied back to the desktop (for example for editing).
- Files can be deleted individually. The file itself and the database entry are deleted. The remaining entries of the corresponding record stay untouched.
- Duplicates within the same record are removed; existing files with the same name are overwritten (handy when you, for example, want to edit a text file, change it, and save it again).

On editing (for example to update a record)

Double-clicking a *field of the desired record in the table* at the bottom of the form loads all entries of that record into the corresponding input fields. There they can be changed, supplemented, extended, and saved again by clicking *Update – change data*. This is not possible for all fields, because otherwise the database would become inconsistent (the calculations would no longer be correct). This is the case, for example, for the entire upper part of Daily leg. These fields have a gray background and are not writable.

There's a special case for the additional modules: their entries are also edited within those modules. There, double-clicking a *list entry* or the photo opens a submenu. This offers the option to Open, save to desktop, or delete by clicking the corresponding buttons.

Opening depends very much on the respective computer and its default applications (for example whether there's any program at all to open .gpx or .kmz navigation files). Double-clicking a PDF should open the document in the computer's default PDF viewer; a Word document (.doc, .docx) in the corresponding office software.

All this sounds complicated because I described all the additional possibilities. If you stick to just the required entries, it's actually quite manageable, and such an entry is also done quickly. The additional possibilities you'll get the hang of with practice. Plus, thanks to the *edit function* entries can be supplemented and changed at any time.

Travel expenses

Simpler, because less extensive, is the Travel expenses form.

Required entries are again marked, and without entries there it cannot be saved (corresponding warnings).

- **Location:** Selection or entry are possible.
- **Cost category:** Only selection possible. To extend: Start menu → Edit selection menus. Only entries can be added, none deleted.

Cost module — cost entry

The same module also appears in the Fueling form. Required entries are *amount and currency*, upon which the corresponding *conversion factor* and the *amount in euros* appear. *Payment method* is optional (selection only, no entry).

A few currencies are already preset. If the matching currency is not among them, or if the conversion factor is no longer correct (for example due to galloping inflation in the host country), this can be corrected by *conversion factor* no longer matches (for example due to galloping inflation in the host country), this can be corrected via Open currency form. There are two options:

- *Create a new currency*, if the currency is not yet stored. There you enter the name of the new currency and the conversion factor. Clicking *Transfer values to form* converts the current amount into euros using the new currency, and also stores the new currency in the Currency selection field for future use.
- *Change the conversion factor of an existing currency:* If the conversion factor is no longer correct, it can be corrected here. *Select currency* (selection only), *enter new factor* and click *Transfer values to form*: from now on the new factor is used for current and future calculations.

Submit data saves the entries to the database, where they appear in the table. This section is present in roughly the same form in almost all forms.

New record clears the input fields (no effect on the database or the table) for entering a new record. *Back to start page* closes the corresponding form and returns to the start page.

Double-clicking a *field of a record* in the table loads that record into the input fields for editing in most forms. As a result, the appearance of the section with

Submit data changes: this button turns gray and is no longer clickable; instead, the buttons *Update – change data* and (not in all forms) *Delete record* appear.

Update overwrites the edited record with the changed values and saves it to the database. *Delete* removes the entire record from the database (so caution, despite the safety prompt).

After *Update* or *Delete*, the empty original form mask appears again. *Submit data* is visible again, *Update – change data* and *Delete record* are hidden.

Measures — conversion

A small form for converting US units to metric and vice versa. Operation is intuitive: *enter value, pick unit from selection menu, click Start calculation*. The calculated values are not saved.

Create new trip

A short but important form. First, as the name says, here a new trip can be created. *Trip*, or rather *General – no trip*, is an important criterion for categorizing the data.

A trip can also be deleted.

Attention: However, this also deletes all records related to that trip — except fueling: that is, travel expenses, daily legs, contacts. Fueling is kept, because otherwise all previous consumption calculations would be void. So use the delete function with extreme caution.

Fueling

Likewise an extensive module, even though it is reached via just one button. In principle there are six sub-units, but they are all operated through the same form, namely:

- Fuel — general (no trip)
- Fuel — during a trip
- LPG — general
- LPG — during a trip
- AdBlue — general
- AdBlue — during a trip

Which fueling types are available depends on the active vehicle. If the vehicle has no AdBlue tank, for example, that selection is not offered.

A fueling log only makes sense if the data is entered regularly and continuously, otherwise the calculations become very arbitrary.

- First, in the selection menu under *Select trip* , pick the matching entry
- Then enter the actual *fueling data*. After selecting What is being fueled, the matching remaining fields appear. All four fields must be filled in; the liter value is calculated. *Full tank* is also important, because consumption calculations only make sense from full tank to full tank. The two remaining fields are optional.
- Then enter the *date* and the *kilometer reading* — in the two gray fields, the data of the last fueling entry is shown. In the white fields below, the *current date* (the last fueling date is preselected) and the *current kilometer reading* are entered. Both entries are mandatory; the program will point this out later if you forget. Furthermore, *warnings* appear when date and kilometer reading are identical to the last entry, when the entered date is before the last entry, when the kilometer reading is smaller than the last entry, or when it is disproportionately much larger. With these *warnings*, saving is still possible (handy if you want to backfill earlier data).
- A special feature in this module is *Activate backfill mode*. This is useful if, for example, you want to transfer a whole bunch of fueling data from a paper fueling log into the app, because in that mode the plausibility warnings are disabled
- The cost module was already discussed.
- Before *submitting the data* , the program checks once more whether the most important entries are consistent. Missing required entries prevent saving.
- This time there are *three tables*: one for *fuel*, one for *LPG* and one for *AdBlue*. Double-clicking an entry loads the data for editing again. With the toggle buttons above the tables, you can show or hide individual tables to keep an overview.

Attention: Be careful when editing or deleting: when important data is changed (for example *date*, *kilometer reading*, *fuel quantity*), all the calculations no longer match. The edit and update function is intended mainly to correct earlier incorrect entries or to supplement optional data. Even more caution when *deleting*.

Special cases when fueling

When fueling, two situations can arise that don't fit the normal scheme: you have no cost information (lost the receipt), or you forgot to record one or more fuelings in between. For both cases, the fueling form has its own small section "Special cases (rare)" with two checkboxes. You'll find it directly below the

Backfill mode banner.

The two checkboxes are independent of each other — you can set one, both, or none.

Cost unknown

You did a fueling but no longer have the receipt — or you're backfilling old fuelings from a paper fueling log where no price was noted. Then you tick the "Cost unknown" checkbox.

What happens then:

- The fields Amount, Currency, Factor, Euro, and Payment method are locked and shown in gray. You don't have to (and can't) enter anything there.
- The fueling is still saved as normal — with date, kilometer reading, fuel quantity, and everything else.
- In the fueling tables, the Euro column shows a dash "—" in gray, so it's visible at a glance: the cost information is intentionally missing here.
- In the consumption calculation, the fueling is included in the liter calculation (l/100 km) as normal — it is only missing from the cost calculation (€/l, total cost).

This way your consumption calculation in liters stays correct even if you occasionally have no cost data. Only the euro total is then incomplete — and the program tells you in a hint after the calculation how many fuelings without cost information were involved.

Restart fueling sequence from here

More tricky is the case where you have completely forgotten one or more fuelings in between — date, kilometers, and liters are all gone. The fueling sequence is then broken: from the last fueling before the gap to the next one after, the liters-per-100-km values no longer match, because the forgotten fuelings in between are missing arithmetically. The program would otherwise calculate a much too low consumption.

For this case, there is the checkbox "Restart fueling sequence from here". You set it on the first fueling AFTER the gap. With this you tell the program: forget everything that came before, start fresh from here.

What happens then:

- The fueling itself is saved as normal with all data.
- In the fueling tables, the row is highlighted with a light blue background, so you can see at a glance: this is a sequence restart.
- In future consumption calculations, all fuelings BEFORE this entry are

no longer included in the same sequence. The consumption calculation starts fresh from the sequence restart.

Before the checkbox is actually set, the program asks again with a confirmation dialog. If consumption calculations from the affected period are already saved, it also tells you how many — those could become outdated by your marking, and you may need to recalculate them.

Attention: The checkbox "Restart fueling sequence from here" exists only for fuel (diesel/gasoline) and AdBlue. For LPG it is hidden, because LPG fuelings are not part of a continuous kilometer sequence anyway — for LPG it is only about date, quantity, and cost.

Both special cases when editing

When you open an already-saved fueling for editing with a double click, both checkboxes are automatically set to the state they had when saved. You can change them and save the update.

Especially handy is this when adding cost data later: you saved a fueling as "Cost unknown" because you didn't have the receipt, and you find it later in your pocket. Then you open the fueling for editing, uncheck the box, enter amount and currency, click "Update – change data" — done. From now on the fueling is again fully included in cost calculations.

It works the other way too: if you later realize that the cost information back then was wrong and is no longer reconstructible, you can set the "Cost unknown" checkbox during editing. The old cost data is deleted on update — the program asks again first for safety.

Attention: If you uncheck the "Cost unknown" checkbox during editing, you **MUST** enter amount and currency, otherwise the program refuses the update. If you change your mind, you can simply click the checkbox again.

What if I want to backfill a completely forgotten fueling

Sometimes you remember after the fact all the data of the forgotten fueling (date, kilometers, liters, possibly cost). Then you don't need a sequence restart, you simply backfill the fueling — even right in the middle of your existing fueling log.

That's what Backfill mode is for. Activate it via the button in the Backfill banner and then enter the missing fueling with its original date and kilometer reading. The program suppresses the soft date/km warnings that would otherwise appear ("Date is before last entry" and similar).

One important safeguard remains active, though: the Corridor check verifies that your kilometer reading fits between the previous and following entries. If you backfill, for example, between a fueling at 50,000 km and one at 51,000

km, your entered kilometer reading must lie between these values — otherwise a clear warning appears. This protects against accidental typos and broken consumption sequences.

After backfilling, the consumption calculation automatically includes the new entry in future calculations — it sits chronologically correctly between the others. You don't need to do anything else. You can then deactivate Backfill mode again, or simply close the module — it will be off automatically the next time it is opened.

Consumption calculation

Interesting when fueling is the option for consumption and cost calculation. You reach it via *Start menu* → *Consumption calculation*.

- First, again select a *trip*, as before.
- Next, the *fuel type*.
- Finally, the *date from – to*. Note that only dates on which you filled up are offered (except for LPG). Of course, the "not full tank" fuelings in between are included in the calculation.
- After clicking *Start calculation*, a table is first shown with all selected fueling events.
- Below that comes the actual consumption calculation. These calculations are saved. The bottom table shows all previous calculations with a corresponding delete function (just one row, all previous calculations).

So you can have it calculated, for example: how much diesel did I refuel on my France trip from 02/03/2022 to 08/04/2022, what was the average consumption, and how much did it cost.

Edit selection menus

Here the preset selection lists can be extended — for example adding new *cost categories*, *travel expense categories*, or *fueling categories* if the presets don't fit. Deletion of entries is intentionally not possible here, because otherwise records in the database would lose their references and become invalid.

Database queries

The big advantage of a relational database is that you can link different data and then output the result. For example: how much did I spend on food on trip XXX between dates YYY and ZZZ. Or: what was my fuel consumption between YYY and ZZZ, and how much did it cost.

Of course, the precondition is that you maintain the database and enter all the

data. What's not in there can't be calculated. Data can't be changed here, that's only possible in the corresponding menus.

You reach the form *Database queries* via the *start page* in the *Other section*. It is divided into several tabs:

Filter by trip and date

In the upper area of the form, you can pick a *trip*. Then in the *date selection fields*, only those dates are offered for which entries actually exist within the selected trip. So you don't have to guess which dates are available in the database.

With *Apply date filter*, the tables below are restricted to the chosen period. With *Reset filter*, the date filter is removed and you see all entries of the trip. Handy if you, for example, only want to see the data from the first travel days of a long trip.

Below the filtered tables, *totals* are shown: for the daily legs the *total of daily kilometers*, for travel expenses the *total in euros*. This gives you a direct overview of how many kilometers you drove, for example, in the first travel week, or how much money a particular region cost you.

The individual tabs

- **Contacts:** All stored contacts categorized by various criteria: trip, first name, last name.
- **Daily leg:** The travel diary, ordered by trip and optionally narrowed by the date filter.
- **Travel expenses:** Expenses on the road (except fueling), again ordered by trip and narrowable with the date filter. Fueling can be viewed via the consumption calculation.
- **Notes:** Stored notes (PDFs, DOCX etc.) of the selected trip. Double-clicking an entry opens the file in your computer's default program.

Backup and restore

No matter how carefully you work — a database can be lost through hardware failure, software errors, or accidental deletion. That's why there is a separate module for Backup and Restore, reachable via *Start page* → *Other* → *Backup / Restore*.

What is backed up

A backup includes:

- The *complete database file* of the currently active vehicle

- The *entire PDF folder* with all photos, notes, and navigation files
- A small info file with date and time of the backup

This contains everything needed for a complete restore point.

Create backup

The module shows you the *current backup path*. By default, this is a directory in your operating system's user data folder. You can use *Change path* to choose a different location — for example an external USB stick, a cloud-synchronized folder, or a network drive. The chosen path is saved and applies to future backups.

Clicking *Create backup now* starts the backup. A progress bar shows the progress, and you can stop the operation at any time via *Cancel*.

So that storage doesn't grow endlessly, at most the *last three backups* are kept — older ones are automatically deleted with each new backup. If you want to keep more versions, you should copy individual backup directories elsewhere.

Restore — restoration

From the *list of existing backups*, you can pick one and restore it with *Restore*. There are three built-in safety stages before anything is actually overwritten:

- Before the restore, a *pre-restore backup of the currently active database* is automatically created — so if something is wrong after the restore, you can still go back.
- The backup database to be imported is checked for plausibility (at least one vehicle present, reference tables not empty).
- A *confirmation dialog* shows once more what will happen and asks explicitly.

Only when all three stages pass is the restore actually carried out.

Attention: After a successful restore, *the program automatically exits*. This is intentional: the running database connection still points to the old file after the overwrite — only a restart of the program loads the restored database cleanly. *Just restart the program*, and you'll be working with the restored state.

Pre-restore backups

The *pre-restore backups*, which the program automatically creates before each restore, are stored not in the normal backup directory but in a separate subfolder. They are also not subject to the "max three" rotation — they accumulate, because they are intended as a last lifeline. If after years it becomes too many for you, you can clean up the pre-restore folder manually.

Trip download

Sometimes you want to share the data of a trip with others — for example pass on the tracks of the Italy trip to a friend or upload the photos of the France trip to the family cloud. That's what the Trip download module is for, reachable via Start page → Other → Trip download.

Unlike a backup, which always saves everything, the download is selective: you pick a trip and decide which file types you want to export — photos, tracks/navigation files, notes, or a combination of these.

Operation

- Pick a trip — the list shows only real trips, so not "General – no trip" or "Other".
- Tick file types — all three are initially activated. You can also export, for example, only photos or only tracks.
- A statistics line shows how many files are selected in total and how much storage that comprises — handy to estimate before the download whether the target (for example a USB stick) has enough space.
- Clicking Start download opens a directory selection dialog. Pick the desired target folder.
- In the target folder, a new subfolder with the trip name is created, and within it, depending on selection, the subfolders Photos, Navigation, and Notes with the corresponding files.

What if something is already there

If a directory with the trip name already exists in the target folder (for example because you've exported once before), the program automatically appends a number: Italy_2, Italy_3, and so on. So you can't accidentally overwrite anything.

If a single file is missing or cannot be copied, this is noted in the final result dialog — the rest is still copied. So you always get everything that is copyable, and still see where something didn't work.

Closing words

That's essentially it. Womo-Reise has several modules, and on first reading it may seem like a lot — but in everyday use you rarely need everything at once. A typical travel day for entries looks like this: in the evening, briefly fill in the daily leg, possibly attach a photo or a track; when fueling, record a fueling; when shopping, enter the travel expenses. With this you've essentially covered the daily routine.

Backup you do every now and then, maybe every few travel days. Database queries and Trip download you use when you want to know or share something specific.

Have fun with it!