TribeManager User Manual

Table of Contents

This manual is for people who **use** TribeManager day to day: managing contacts, planning events, working through tasks, and so on.

It is ordered by complexity mode (see SPEC.md, *Module complexity modes*): basic, advanced, then expert. Each mode builds on the one before it, so read only as far as the mode your instance is configured for.

1. Basic functions

Basic mode is the bare minimum for each module to work, make sense, and link to other entities: a task has a status, an event has a time, an inventory item just exists. No groups, categories, templates, or planning surfaces.

1.1. Contacts

Add a person from the **Persons** screen with **New person**: in basic mode a contact needs only a **name**. Open a person to **edit** their details, or **delete** them. Use the **search box** to find people by name or any other field.

Long lists are shown a page at a time. Below the list, **First page**, **Previous page**, and **Next page** move through the result set, and the line between them says which page you are on and how many records matched altogether. Above the list, **Sort by** chooses the field to order by and **Direction** whether it runs from A to Z or back. Changing the sort always returns you to the first page - page 7 of a different order would be nowhere in particular. Records that do not have the field you sorted by are listed at the end, in both directions.

Nothing you enter is ever rejected. If something is incomplete or does not look right - a missing name, a date that is not a date - it is still saved and shown as a **warning** so it can be fixed later (the general rule everywhere: warn, don’t block).

1.2. Events

Add an appointment from the **Events** screen with **New event**: an event needs a **title**, a **status**, and a **start**. Everything else is optional.

All day switches between the two kinds of date an event can have. A whole-day event carries just a date; unticking **All day** adds a time and a time zone, which is stored with the event so a change to daylight-saving rules never moves it. The **end** may be left empty for an event without a fixed finish.

The **status** says where an event stands: **In planning**, **Confirmed**, or **Cancelled**. The fourth value, **Template**, marks an event as a copy source: it has no date and does not appear in the event list unless you tick **Include templates**.

A **venue** is picked from the organization’s location contacts, and events can be nested - a sub-event is a full event with its own time and place, which is how a festival splits into setup, the event itself, and dismantling. A sub-event that starts before its parent is fine; the tool only points it out.

Filter the list by text, by a date range - an event overlapping the range is included, not only one entirely inside it - and sort it by date, title, or status.

1.2.1. Calendar

The **Calendar** screen shows a month at a time. **Previous month**, **Today**, and **Next month** move through it, and an event lasting several days appears on each of those days rather than only on the first. Selecting an event opens it.

Under the grid, **Subscribe** offers the calendar to your own calendar programme - Thunderbird, Outlook, a phone, anything that can subscribe to a calendar link. Copy the link, paste it into your programme, and it keeps itself up to date: new and changed events appear without anybody sending anything. Templates are left out, since they have no date.

1.2.2. Participants

Open **Participants** next to an event to manage who is taking part. Pick a person, pick a **role**, and add them; unless you untick **Send an invitation**, they receive a real calendar invitation by email with accept and decline buttons in their own mail programme.

Each participant has a **status** - invited, requested, confirmed, declined, tentative, or on the waiting list. **Attended** is a separate tick: someone who confirmed and then did not turn up is not a cancellation, and the two facts are kept apart so the register stays honest.

Places per role limits how many people may confirm in one role. A role without a limit takes as many as sign up. A limit never stops anyone - confirming one person too many is allowed and simply pointed out, because you know your room better than the software does. The same goes for a person without an email address: they join the list, they just do not get an invitation.

A **reminder** is set as a number of minutes before the start, not as a fixed time, so moving the event moves the reminder with it.

Moving an event or changing its venue sends everyone an update; cancelling it sends a cancellation that takes the appointment out of their calendars again. Renaming it sends nothing.

1.3. Tasks

Add a to-do from the **Tasks** screen. A task needs only a **title** - no deadline and no owner - because a to-do you cannot write down until you have decided everything about it is one that ends up on a different piece of paper.

A **deadline** is optional and can be a whole day or an exact time. The **status** moves between **Open**, **In progress**, and **Done**, in any direction: reopening something is a normal thing to do.

1.3.1. Who does it, and who answers for it

These are two different questions and the tool keeps them apart.

Assigned to is who does the work. It can be several people, and it can be an organization - "the caterer handles the food" is a real assignment.

Responsible is who answers for the task. At most one person, and it may be left empty while you are still planning.

If you leave it empty and exactly one person is assigned, that person is responsible - and the screen always says **(via the assignment)**, so a derived answer is never mistaken for a decision somebody made. Hand a task back to that state on purpose by choosing **whoever is assigned**: from then on the responsibility follows the assignment, which is exactly what you want when the job changes

hands.

If nobody is assigned, or several people are, the task says that nobody answers for it - quietly while it is still open, and more insistently once work has started.

1.4. Inventory

The **Inventory** screen lists what the organization owns, how many of each there are, and how many are currently free.

Kept at is one of your location contacts, and **Shelf** is a free-text label. It suggests the labels already used in that room, so spellings stay consistent without anybody having to maintain a list of shelves - and when the last thing leaves shelf 3, shelf 3 simply stops being suggested.

Things can contain other things: put a tool inside a toolbox, and the tool stays a full entry with its own count, borrowable on its own.

1.4.1. Kit definitions

Tick **This is a kit definition** to describe what should be in something - the standard contents of a first-aid kit, say. A definition never appears in the stock list, because it is not a thing you own.

Build a real kit from a definition, and the kit remembers where it came from. From then on it tells you what is missing compared to the definition, and what is in it that the definition does not mention.

1.4.2. Lending out

Lend something out claims a quantity for a person, or for an event. An event's claim uses the event's own dates. A loan may have no expected return date - leave it empty when nobody knows yet.

Record **Handed over** when it actually leaves and **Came back** when it returns. Overdue then works itself out from those dates and shows in the list; there is no button anybody has to remember to press.

Claiming more than there is, or claiming a box whose contents are already out, is **allowed**. The conflict is pointed out and the claim is kept - you know your situation better than a list does.

1.5. Finances

Each **account** - your bank account, the cash box - has its own **cash book**: what happened, in date order, with the balance after each line.

Two figures sit at the top and they mean different things. **Balance** is the money that is actually there, the number a bank statement would show. **With planned entries** is what it becomes once everything you have already recorded as expected has happened. Keeping them apart is the point: one tells you where you stand, the other whether you survive the invoices you already know about.

1.5.1. Entries

Record an entry with an amount, a date, and a **category**. Negative amounts are money going out. Both a dot and a comma work as the decimal mark.

Tick **Expected, not yet happened** to record something that has not occurred yet - a budget line, an invoice you know is coming. It appears in the register and counts towards the outlook, but not towards the balance.

1.5.2. Transfers

Tick **This is a transfer** when you move money between two of your own accounts. A transfer has no category, and that is deliberate: moving your own money is neither income nor an expense, so it never appears in the income statement. Many simple bookkeeping tools get this wrong and count the same money twice.

1.5.3. Income and expenses

The **Income and expenses** section is the EÜR: what came in, what went out, and the difference. Categories are listed with what actually happened, and separately with what is still expected.

An organization keeping accounts in more than one currency gets one statement per currency. Nothing in this software converts between currencies, on purpose - a total built from an invented exchange rate looks authoritative and is not.

1.6. Documents

Upload a file from the **Documents** screen. A name is optional - the file name is used when you leave it empty. Find documents again by name, description, or file name, and download them with the link beside each one.

1.6.1. Versions

Choose an existing document under **New version of**, and the file you upload is added to it as a new version instead of becoming a second document. Nothing is ever overwritten: earlier versions stay, and uploading an older file again simply makes it the current one - history moves forward rather than backwards.

Uploading a file the organization already has stores it only once. Two documents can point at the same content without either of them being a copy.

1.6.2. Attaching

Attach a document connects it to a person or to an event - the membership form to the member, the minutes to the meeting. The connection works from both ends: a person's documents and an event's documents are the same links read the other way round.

Deleting a document removes the entry but leaves the file content in place, because another document may be using it.

The only thing that is refused outright is a file larger than the instance allows. Everything else in this software accepts what you give it and points out what looks wrong; a size limit is different, because the harm would be the storing.

Basic-mode functions for Communication are added here as they land.

2. Advanced functions

Advanced mode adds more possibilities with reasonable presets and a limited, curated set of options - sophisticated but not overengineered, for the everyday user. This chapter covers what advanced mode adds on top of the basic functions.

2.1. Contacts

In advanced mode a contact carries the everyday fields the organization has turned on - name, email, phone, address, birthday, and any custom fields. The contact form shows a matching input for each: a date picker for dates, a dropdown for single-select fields, a plain box for text. Warnings still guide rather than block - an odd or missing value is saved and flagged, never refused.

The **Contacts** screen lists every kind together - people, organizations, and places - so the club's insurance broker is as findable as its members. Narrow the list by kind, and open **Filter by field** for the typed filters: an exact value for text and selection fields, a from/to range for dates and numbers, so "everyone born before 2008" is a query rather than a scroll. Only the fields your organization has turned on are offered.

While you type a name or an email address, TribeManager looks for contacts that already exist - the same address, or a name made of the same words - and links them. It never stops you: if the two really are different people, save and carry on.

Open a contact to manage its **relationships** to the organization - member, customer, supplier, and so on - each with a validity period and, for members, a status (applicant, active, passive, honorary, resigned). A relationship changes as life does: switch the status, or end it - one click for "ends today" - and the relationship stays on the record with its history intact. **Remove** is for the entry you made by mistake, not for the membership that ended. The **Members** screen lists everyone who is currently an active member; membership is derived from these relationships, there is no separate member record.

2.2. Events

Advanced mode is where an event stops being a single appointment. Nest events - setup, the event itself, dismantling - each a full event with its own time, place, and participants, so the setup crew and the audience are genuinely different lists rather than one list with a note.

Use **templates** as copy sources for the things you run every year, and limit **places per role** where a room or a minibus sets the number. A limit never stops anyone: going over it is pointed out, because the person booking usually knows something the software does not.

2.3. Tasks

Hang tasks on an event with a temporal relation - before, during, after - so an event carries its own checklist. Nest tasks the same way events nest.

The distinction between **assigned to** and **responsible** comes into its own here: assign a whole group of people to the work, and name one of them as answering for it - or leave it to follow the assignment, and let the duty move when the job changes hands.

2.4. Inventory

Contain items in other items, and define **kits** whose contents are checked against a definition: what is missing from the first-aid box, and what is in it that should not be. Reserve material for an event and it takes the event's own dates, so the two cannot drift apart.

2.5. Finances

Budget with **planned** entries: they show in the register and in the outlook without touching the balance, and turn into reality when you record what actually happened. Attach entries to an event to see what it cost and what it brought in.

2.6. Communication

Write a message under **Mailings**: a subject, a text, and the kind it is. Saving keeps it; nothing leaves the building until you send it, which is a separate click on the mailing itself. Writing and sending are apart on purpose - a draft should never be something somebody already received.

Write `{recipientName}` into the subject or the text and each person sees their own name there. Anything else you write stays as it is, including a marker nobody knows - a visible `{amount}` in a sent mail is a mistake you can find, an empty gap is not.

Tick **Keep as a template** and the message becomes a copy source instead: it is never sent, and **Use this template** writes a fresh mailing from its words. A mailing that already went out can be written again the same way, which is what you want for last year's invitation.

A sent mailing carries the date it went out and cannot be sent a second time. It reports how many people it reached, how many had declined that kind, how many are waiting for their digest, and how many have no address on file.

2.7. Data protection

Under a contact you can produce two documents. The **erasure report** says what would be anonymized and what statute keeps, and hands the person who asked a readable answer before anything happens. The **access request** is the other half: what is stored about them in full, where it came from, and how many postings, participations, and relationships elsewhere point at them - counted rather than reprinted, so the document is not another copy of the data.

If your organization takes donations, mark the finance category they are booked under (see the admin manual). Then a person's **Zuwendungsbestätigung** for a year is a download: the collective receipt over everything they gave, in the wording German tax law requires, with the amount written out in words. Naming a single posting confirms that one gift instead.

Advanced-mode functions for Documents are added here as they land.

3. Expert functions

Expert mode is the full escalation of configurability - everything the domain model can carry, exposed at once. Powerful and deliberately dense. This chapter covers what expert mode adds on top of the advanced functions.

3.1. What expert mode is for

Expert mode does not add data. Everything an expert-mode organization records is recorded the same way in basic mode - the difference is how much of the surface you are shown. Raising or lowering a module's mode therefore never migrates anything: something created in expert mode simply hides in a lower one, and reappears when you raise it again.

That is worth knowing before you turn it on. If a mode is too high, turn it back down; nothing is lost by trying.

Expert-mode functions per module are added here as they land, following the scope each module's section fixes.

3.2. Contacts

A place is rarely just a place: give a location contact the location it is **part of** and the address book grows a map - campus, building, room. Every node stays a full contact, usable as a venue or a storage place on its own.

Phone numbers and email addresses carry a **type** - work, home, cell, or whatever your organization uses. The list is not fixed: vCard's own registry is open, so what you type is kept and exported as you wrote it.

Record **contact persons** for the organizations you deal with: which individual speaks for which partner, with an optional role such as chair or purchasing. It is a communication pointer, deliberately not a statement of who is responsible for anything - responsibility lives with tasks.