grommunio Command-line Reference

Official documentation

Version 2026-07-02 · docs.grommunio.com

Command-line administration

Almost everything you can do in the **grommunio Admin UI** — and a good deal more — can be driven from the shell on the grommunio server. The command line is the fastest way to perform bulk changes, script recurring jobs, and reach low-level mailbox operations that have no UI equivalent.

There are two families of tools, and it helps to know which layer you are working at.

grommunio-adminThe high-level management CLI. Create and modify users, domains, organizations and mailing lists; configure LDAP/AD and fetchmail; manage passwords, services and configuration. Operates on the same data as the Admin UI / REST API. →
Gromox CLI utilitiesThe low-level tools that act directly on stores and mailboxes — gromox-mbop (mailbox operations), store creation, check/repair, sizing, snapshots and import/export. Use these for maintenance and data-level tasks. →

Which tool do I reach for?

You want to …	Use
Create / modify / delete a user or domain	`grommunio-admin user`, `grommunio-admin domain`
Set or reset a password	`grommunio-admin passwd`
Connect / synchronize an LDAP or Active Directory	`grommunio-admin ldap`
Pull mail from a remote mailbox	`grommunio-admin fetchmail`
Empty a folder, delete messages, purge soft-deleted items	`gromox-mbop`
Reclaim disk space in a store	`gromox-mbop` (`vacuum`, `purge-datafiles`)
Check or repair a mailbox	`gromox-mbck`
Report a mailbox's size	`gromox-mbsize`
Run a one-off action across *all* mailboxes	`gromox-mbop foreach.*`

> 🚀 **New here? Start with the cookbook**
>
> The **Common administration tasks** page collects ready-to-use recipes for the jobs administrators run most often — onboarding users, LDAP synchronization, mailbox clean-up, fetchmail and backups.

Conventions

Run these commands on the grommunio server, as `root` (or with `sudo`). A few conventions recur throughout the reference:

- **`USERSPEC`** / **`MBSPEC`** — a user is identified by e-mail address (`jdoe@example.com`); some commands also accept a numeric ID.
- **`grommunio-admin <area> <action>`** — the management CLI is organized into areas (`user`, `domain`, `ldap`, …), each with its own actions (`create`, `modify`, `list`, `delete`, …). Append `--help` at any level.
- **`gromox-mbop -u <mailbox> <action>`** — the mailbox-operations tool always takes the target mailbox with `-u`, followed by one or more actions.
- Destructive commands are flagged with a caution admonition — read those before running them in production.

Offline reading

This section is also available to download for offline use:

- Download as PDF
- Download as EPUB

Common administration tasks

A practical cookbook for the jobs administrators run most often. Every command is run on the grommunio server as `root` (or with `sudo`). Identify a user by e-mail address (`jdoe@example.com`); append `--help` to any `grommunio-admin` command to see all its options.

> **Note**
>
> These recipes pair the high-level `grommunio-admin` management CLI with the low-level `gromox-*` mailbox tools. For the full option set of any command, follow the links to its reference page.

Users & passwords

Create a user and set a password

```
# Create the mailbox (a maildir/store is provisioned automatically)
grommunio-admin user create jdoe@example.com

# Set an initial password (you'll be prompted), or generate a strong one:
grommunio-admin passwd jdoe@example.com
grommunio-admin passwd -a -l 16 jdoe@example.com     # auto-generate, 16 chars
```

The domain (`example.com`) must already exist — see Create a domain below.

Inspect, list and search users

```
grommunio-admin user show jdoe@example.com           # full detail for one user
grommunio-admin user list                            # all users
grommunio-admin user list -f status=0 jdoe@*         # filter + wildcard
```

Modify a user

```
# Add an alias address
grommunio-admin user modify jdoe@example.com --alias john.doe@example.com

# Change the interface language
grommunio-admin user modify jdoe@example.com --lang en_US

# Toggle a feature (e.g. disable ActiveSync for this user)
grommunio-admin user modify jdoe@example.com --privEas 0
```

`grommunio-admin user modify --help` lists every field, including the per-user feature switches (`--privChat`, `--privVideo`, `--privFiles`, `--privDav`, `--privEas`, …), aliases and stored properties.

Manage mobile devices

```
grommunio-admin user devices jdoe@example.com list           # paired EAS devices
grommunio-admin user devices jdoe@example.com resync DEVICE  # force a resync
grommunio-admin user devices jdoe@example.com wipe DEVICE    # remote wipe
```

Delegation and send-as

```
grommunio-admin user delegate jdoe@example.com add assistant@example.com
grommunio-admin user sendas   jdoe@example.com add shared@example.com
grommunio-admin user delegate jdoe@example.com list
```

Delete a user

> ⚠ **Destructive**
>
> Deleting a user removes the mailbox. Add `-k` to keep the underlying files on disk if you may need to recover them.

```
grommunio-admin user delete -y jdoe@example.com      # -y skips the confirmation
```

Domains & organizations

Create a domain

```
# -u sets the maximum number of users for the domain
grommunio-admin domain create -u 100 example.com
```

List, modify and remove domains

```
grommunio-admin domain list
grommunio-admin domain modify example.com            # see --help for fields
grommunio-admin domain delete example.com            # soft-delete (recoverable)
grommunio-admin domain purge --files example.com     # permanent + remove files
```

> **⚠Destructive**
>
> `domain purge --files` permanently deletes the domain **and** all of its mailboxes' data. There is no undo.

LDAP / Active Directory

Connect grommunio to an external directory, then import and keep users in sync.

```
grommunio-admin ldap configure        # interactive: server, bind, search base
grommunio-admin ldap check            # verify connectivity and the configuration
grommunio-admin ldap search jdoe      # find matching directory objects
grommunio-admin ldap dump jdoe@example.com   # show the raw LDAP object
```

Import users (a "down-sync" from the directory into grommunio):

```
grommunio-admin ldap downsync jdoe@example.com   # one user
grommunio-admin ldap downsync -c                 # complete sync of all mapped users
```

> **🚀Automate it**
>
> Run a periodic `grommunio-admin ldap downsync -c` from a systemd timer or cron job to keep grommunio aligned with the directory. `grommunio-admin ldap check -r` reports (and, with `-y`, removes) users whose directory object has disappeared.

Fetchmail — pull mail from a remote mailbox

Useful during migrations to collect mail from a user's old provider:

```
grommunio-admin fetchmail create \
  --srcServer mail.old-provider.example \
  --srcUser   old-account \
  --srcPassword 'secret' \
  jdoe@example.com

grommunio-admin fetchmail list jdoe@example.com
```

Mailbox maintenance

These tasks use `gromox-mbop` ("mailbox operations"), which always targets a mailbox with `-u`. Folders can be given by **symbolic name** (`INBOX`, `SENT`, `DRAFT`, `JUNK`, `DELETED`, …) or by path (`/Top of Information Store/…`).

Empty a folder

```
# Empty the Junk folder (soft-delete, like a client would)
gromox-mbop -u jdoe@example.com emptyfld --soft JUNK

# Empty Deleted Items recursively, including subfolders (-R), hard delete
gromox-mbop -u jdoe@example.com emptyfld -R DELETED
```

Delete specific messages

```
gromox-mbop -u jdoe@example.com delmsg -f INBOX 0x1234 0x1235
```

Reclaim disk space

Soft-deleted ("recoverable") items still occupy space until purged. A typical clean-up, then a compaction, frees the most:

```
# Hard-delete items soft-deleted more than 30 days ago, across the whole store
gromox-mbop -u jdoe@example.com purge-softdelete -r -t 30d /

# Drop attachment/content files no longer referenced by any message
gromox-mbop -u jdoe@example.com purge-datafiles

# Compact the SQLite store
gromox-mbop -u jdoe@example.com vacuum
```

Recompute the reported store size

```
gromox-mbop -u jdoe@example.com recalc-sizes
```

Run an action across every mailbox

The `foreach.*` pseudo-command applies an action to many mailboxes at once — ideal for fleet-wide maintenance:

```
# Purge old soft-deleted items in every mailbox hosted on this server
gromox-mbop foreach.mb.here purge-softdelete -r -t 30d /
```

Backups & snapshots

grommunio appliances take periodic, space-efficient snapshots of the mailbox storage (on a copy-on-write filesystem such as Btrfs):

```
/usr/libexec/gromox/gromox-snapshot       # create a snapshot now
```

Schedule it from a systemd timer for regular, low-overhead point-in-time backups. See `gromox-snapshot` for retention details.

Diagnostics & troubleshooting

```
# Open (touch) a mailbox to confirm the store responds
gromox-mbop -u jdoe@example.com ping

# Check a mailbox for inconsistencies (and repair with care)
gromox-mbck jdoe@example.com

# Report a mailbox's size breakdown
gromox-mbsize jdoe@example.com

# Inspect the local delivery queue
gromox-mailq

# Drop into an interactive admin shell (REST API context)
grommunio-admin shell
```

🚀 Keep going

This is a starting set — see the full `grommunio-admin` reference for every management area, and Gromox CLI utilities for the complete list of low-level tools.

Gromox CLI utilities

Where `grommunio-admin` manages the *management* layer (users, domains, configuration), the **`gromox-*`** utilities act directly on the storage layer — the per-user **stores** (`exchange.sqlite3` plus the attachment/content files). They live under `/usr/sbin` and `/usr/libexec/gromox/` and are the right tools for maintenance, repair and data-level migration.

> **⚠Caution**
>
> These tools operate on live mailbox data. Several are destructive. Take a snapshot or backup before bulk or repair operations, and prefer running them during quiet hours.

gromox-mbop

`gromox-mbop` ("mailbox operations") is the workhorse for per-mailbox actions. It always selects a target mailbox first, then performs one or more actions:

```
gromox-mbop -u jdoe@example.com <action> [args...]
```

See the `gromox-mbop(8)` manual page for every action and flag. The actions most relevant to administrators:

Action	What it does
`emptyfld`	Empty one or more folders. `--soft` mimics a client delete; `-R` recurses into subfolders; `-t age` limits to items older than *age*.
`delmsg`	Delete specific messages from a folder (`-f folder`) by message ID.
`purge-softdelete`	Hard-delete soft-deleted ("recoverable") items. `-r` recurses; `-t timespec` limits by age.
`purge-datafiles`	Remove attachment/content files on disk no longer referenced by any message.
`vacuum`	Compact the store's SQLite database (`VACUUM`).
`recalc-sizes`	Recompute the store's reported size.
`ping`	Open the mailbox on the server — a quick "is the store healthy?" check.
`unload`	Drop the store from the server's in-memory cache.
`set-locale`	Set the mailbox locale (drives the translated names of built-in folders).
`get-photo` / `set-photo` / `clear-photo`	Read, set or remove the user's contact picture.

Action	What it does
`get-websettings` / `set-websettings`	Read or write the per-user grommunio Web settings (JSON).
`sync-midb`	Rebuild the midb index used by the IMAP/POP front-ends.
`clear-rwz`	Remove the cached Outlook rules organizer (`IPM.RuleOrganizer`) messages.
`clear-profile`	Clear the cached PHP-MAPI profile for the store.
`cgkreset`	Reset change numbers/keys on all objects — a recovery step for sync corruption.

Folder specifications

Folders are addressed either by a **symbolic name** (resistant to language settings) or by **path**:

```
gromox-mbop -u jdoe@example.com emptyfld --soft JUNK
gromox-mbop -u jdoe@example.com emptyfld "/Top of Information Store/Archive/2022"
```

Recognized symbolic names include `INBOX` , `SENT` , `DRAFT` , `OUTBOX` , `JUNK` , `DELETED` (a.k.a. `TRASH` / `WASTEBASKET`), `CALENDAR` , `CONTACTS` , `TASKS` , `NOTES` , `JOURNAL` and `IPM_SUBTREE` . The slash is always a hierarchy separator (symbolic names work on private stores only).

Command chaining

Multiple actions can be chained for one mailbox by wrapping each in parentheses:

```
gromox-mbop -u jdoe@example.com ( purge-softdelete -r / ) ( purge-datafiles )
```

Run an action across every mailbox

The `foreach.*` pseudo-command repeats an action over many mailboxes — perfect for fleet-wide maintenance:

```
# Every mailbox hosted on this server
gromox-mbop foreach.mb.here purge-softdelete -r -t 30d /
```

Store creation

Stores are normally provisioned for you by `grommunio-admin user create` . The underlying primitives are available directly when needed:

Tool	Purpose
`gromox-mkprivate(8)`	Create a blank **private** (per-user) store.
`gromox-mkpublic(8)`	Create a blank **public** (per-domain) store.

Maintenance & inspection

gromox-mbck

Check a mailbox for structural inconsistencies and, with care, repair them.

```
gromox-mbck jdoe@example.com
```

See `gromox-mbck(8)`. Run on a quiesced mailbox and take a snapshot first.

gromox-mbsize

Report a mailbox's size and what is consuming it — handy for quota investigations.

```
gromox-mbsize jdoe@example.com
```

See `gromox-mbsize(8)`.

gromox-dbop

User-database maintenance for the management database (schema upgrades and related operations). See `gromox-dbop(8)`.

gromox-mailq

List the local delivery agent's queue — useful when mail appears stuck.

```
gromox-mailq
```

See `gromox-mailq(8)`.

gromox-snapshot

Create a space-efficient, point-in-time snapshot of the mailbox storage on a copy-on-write filesystem (e.g. Btrfs):

```
/usr/libexec/gromox/gromox-snapshot
```

Schedule it from a systemd timer for low-overhead, regular backups. See `gromox-snapshot(8)`.

Import & export

For migrations and data recovery, Gromox can move messages between mailboxes and on-disk formats. The conversion tools form a small pipeline — for example `gromox-eml2mbox(8)` and `gromox-mbox2mt(8)` bridge RFC 5322 / mbox data and the internal "mail transfer" format that loads into a store. The Mailbox transfer format page describes the format in detail; browse the full set on the Man Pages index.