Shared Mailboxes

Shared Mailboxes

Admin path: **Email Server > Shared Mailboxes** (`view_shared_mailboxes.cfm`, `inc/shared_mailbox_actions.cfm`, `inc/sync_shared_mailbox_acl_file.cfm`, `inc/sync_user_folder_acl_file.cfm`, `inc/get_shared_mailbox_permissions_json.cfm`).

This page manages **mailboxes that several users can read from and write to** — typically role addresses like `info@`, `support@`, or `sales@`. A shared mailbox is a real Dovecot mailbox in its own Maildir, but it has no login of its own; users access it through their own credentials and the rights granted on this page. The **master switch** for the entire shared-mailbox feature lives on Email Server > Settings (Mailbox Sharing card) — when that switch is off, the rows on this page are preserved but inactive, and the Add / Manage Permissions / Rebuild buttons are disabled.

Per-member rights are stored in the `shared_mailbox_permissions` table and projected to Dovecot's on-disk `dovecot-acl` files via the vfile driver, which is the only per-mailbox ACL driver shipped with Dovecot 2.4 (the SQL rights driver was a non-upstream Hermes carry that was removed in the 2.4 rewrite).

How a shared mailbox is wired

A shared mailbox is more than just an ACL — six tables and a Maildir are stitched together on creation:

Component	Storage	Role
Mailbox row	`mailboxes` with `mailbox_type = 'shared'`	Gives Dovecot a userdb entry so the mailbox has a quota, a Maildir, and a sender identity
Shared mailbox row	`shared_mailboxes`	UI metadata: address, display name, auto-subscribe flag, owning domain
Per-member rights	`shared_mailbox_permissions`	Authoritative permission matrix per (shared mailbox, user mailbox) pair

Component	Storage	Role
On-disk ACL	`/srv/mail/<domain>/<local>/dovecot-acl`	Dovecot vfile driver enforcement file — projected from `shared_mailbox_permissions`
Shared namespace visibility	`dovecot_acl_shared` (`acl_sharing_map`)	Tells Dovecot's `Shared/` namespace which users should see this mailbox in their folder list
Recipient policy	`recipients` (Amavis SVF policy + `recipient_type = 'shared'`)	Allows mail addressed to the shared address to pass the Amavis recipient gate
Sender identity	`sender_login_maps`	Lets the shared address be used as a From: by itself (anchor row) and by each member with Send-As granted
Maildir	`/srv/mail/<domain>/<local>/`	The actual on-disk message store. Bootstrapped via `doveadm mailbox create -u <addr> INBOX` so members see it immediately rather than waiting for first delivery

The add handler creates all of these in a single `cftry` block. If any step fails the catch sets `session.m = 30` and the operation fails-loud rather than leaving a partial mailbox.

Permission model — seven flags, projected to IMAP ACL letters

The UI surfaces seven permission flags. Six are IMAP ACL rights enforced by Dovecot; one (Send-As) is a Postfix sender-identity grant.

UI flag	DB column	Dovecot vfile rights	IMAP ACL meaning
Read	`can_read`	`lrs`	`lookup` (see mailbox), `read` (read messages), `write-seen` (set/clear \Seen flag)
Write	`can_write`	`wt`	`write` (set/clear flags except \Seen and \Deleted), `write-deleted` (set/clear \Deleted)
Delete	`can_delete`	`e`	`expunge` (permanently remove messages)
Insert	`can_insert`	`i`	`insert` (append/copy messages into mailbox)

UI flag	DB column	Dovecot vfile rights	IMAP ACL meaning
Post	`can_post`	`p`	`post` (submit messages via the post address — rarely used)
Admin	`can_admin`	`a`	`admin` (modify the ACL itself from an IMAP client)
Send-As	`send_as`	—	Inserts `(sender = shared, login_user = member)` into `sender_login_maps` so the member can use the shared address as From:

The vfile letters are concatenated into a single token per user (e.g., `lrswtie` for read+write+delete+insert). Dovecot 2.4's vfile parser reads each character as a separate right, so the full-word form (`lookup read write-seen ...`) does NOT work — the parser would treat `o` in `lookup` as an unknown right. The `sync_shared_mailbox_acl_file.cfm` include knows this and emits the single-letter form.

The `dovecot_acl` SQL table is still written by the action handlers for legacy/audit reasons, but Dovecot 2.4 no longer reads it. `sync_shared_mailbox_acl_file.cfm` writes the on-disk file every time permissions change, and the **Rebuild ACL Files** button on the page regenerates every file from scratch — used after upgrading to a new Dovecot release or when an admin reports a member can't see a mailbox they should have rights on.

How a save propagates

```
Add Shared Mailbox  ──► shared_mailbox_actions.cfm (add_shared_mailbox)
                                │
                                │  1. Feature guard (Mailbox Sharing = enabled)
                                │  2. Validate prefix + domain + display name + quota
                                │  3. Four-way conflict check
                                │     (recipients, mailboxes, mailbox_aliases,
                                │      virtual_recipients)
                                │  4. INSERT into recipients (Amavis SVF policy)
                                │     + maddr (Amavis address tracking)
                                │  5. INSERT into mailboxes (mailbox_type='shared')
                                │  6. INSERT into shared_mailboxes
                                │  7. INSERT into sender_login_maps (anchor row)
                                │  8. docker exec hermes_dovecot doveadm mailbox
                                │     create -u <addr> INBOX  (bootstrap Maildir)
                                │  9. For each initial member:
```

```
                        |     - INSERT shared_mailbox_permissions
                        |     - INSERT dovecot_acl (legacy)
                        |     - INSERT dovecot_acl_shared (namespace)
                        |     - INSERT sender_login_maps if Send-As
                        | 10. cfinclude sync_shared_mailbox_acl_file.cfm
                        |     → writes /srv/mail/<dom>/<local>/dovecot-acl
                        |       via temp shell script + docker exec -i
                        |       (heredoc pattern; vmail:vmail 0660)
                        v
                cflocation → session.m = 1
```

Add / Edit / Remove permission flows follow the same shape but only touch the rows for one member, then re-call `sync_shared_mailbox_acl_file.cfm` to rebuild that mailbox's `dovecot-acl` file in place. The sync include uses the **temp shell script + heredoc + `docker exec -i`** pattern (it has to — Lucee `cfexecute` argument quoting can't reliably ship multiline content with embedded special characters through `docker exec`).

Cards and modals on the page

Add Shared Mailbox modal

Field	Notes
Domain	Dropdown of mailbox-type domains (`domains.type = 'mailbox'`). The Address Prefix suffix updates live to show the full address.
Address Prefix	Local-part of the email. Validated against `^[a-z0-9._-]+$` — only lowercase letters, digits, dots, hyphens, underscores.
Display Name	Free-form text shown as the mailbox's `name` and in the table. Required.
Quota (GB)	Mailbox quota. Accepts decimals (e.g., `0.5`). Stored as bytes via `Round(quota_gb * 1024^3)`.
Auto-Subscribe	When `Yes` (default), the shared mailbox appears automatically in each member's IMAP folder list. When `No`, members have to manually subscribe to `Shared/<address>` in their client.
Initial Members	Checkbox list of user mailboxes in the selected domain (filtered live as the Domain dropdown changes). Optional — you can grant access later.

Field	Notes
Default Permissions	Seven checkboxes applied uniformly to every selected initial member. Defaults are Read + Write + Insert checked.

The address-prefix suffix and the member-list filter both run client-side when the Domain dropdown changes. Cross-domain members are excluded from the picker even before form submit; the server-side handler re-enforces the same-domain rule with error 26 if a forged post tries to bypass it.

Shared Mailboxes table

DataTables surface — searchable, sortable, paginated, `stateSave: true`.

Column	Source
Actions	Manage Permissions (opens modal) / Delete (opens confirmation modal)
Address	`shared_mailboxes.address`
Display Name	`shared_mailboxes.display_name`
Domain	`domains.domain`
Members	Count of `shared_mailbox_permissions` rows for this shared mailbox
Quota	`mailboxes.quota` divided into GB (1-decimal for whole GB, 2-decimal otherwise)
Auto-Subscribe	YES / NO badge
Status	`Active` (sharing on + mailbox active) / `Inactive` (sharing on + mailbox disabled) / `Inactive (Sharing Off)` (master switch off)

A Domain filter dropdown narrows the visible rows to one domain.

Manage Permissions modal

Opens via the per-row action button. Two sections:

1. **Current Members** — table of every `shared_mailbox_permissions` row for this shared mailbox, with per-right YES/NO badges and Edit / Remove buttons per row. Loaded via AJAX from `get_shared_mailbox_permissions_json.cfm`.
2. **Add Member** — Tom Select user picker (filtered to the same domain as the shared mailbox) + the seven permission checkboxes
   - an Add button.

The Edit Member sub-modal opens on top of the Manage Permissions modal, lets you toggle the seven flags for an existing member, and re-syncs the on-disk ACL file on save. Changes take effect immediately; the member does not need to reconnect their mail client.

Rebuild ACL Files modal

A maintenance action that walks **both** admin-managed shared mailboxes AND user-managed folder shares and regenerates every `dovecot-acl` file from the current state of the database.

> **When to use Rebuild ACL Files.**
>
> - After upgrading to a new Dovecot 2.4 release — backfills the vfile files for any shared mailboxes created before the upgrade.
> - When a member reports they cannot see or access a shared mailbox or shared folder they should have rights on (recovery / drift heal).
> - After manually editing `shared_mailbox_permissions` or `user_folder_shares` in the database.
>
> Safe to run anytime — it rebuilds files from the database and never modifies the permission rows themselves. Per-mailbox failures are non-fatal; the operation continues to the next.

The success banner reports a count of shared mailboxes rebuilt and a separate count of user folder shares rebuilt, so the admin can confirm the operation covered everything they expected.

Delete Shared Mailbox modal

A confirmation modal that lists exactly what will be removed:

- All member permissions and ACL entries
- Sender login maps (send-as permissions)
- Dovecot shared folder subscriptions
- Amavis policy entry

With an optional **Also delete all email messages from the server** checkbox (default checked) that, when set, runs `docker exec hermes_dovecot rm -rf /srv/mail/<domain>/<local>` to remove the Maildir. The DB rows are deleted regardless of that checkbox; only the on-disk messages are conditional. Maildir deletion is wrapped in a non-fatal `cftry` — failure leaves the messages on disk for an admin to clean up later, but the DB state is correct.

User-initiated folder shares — same engine, different page

Individual users can share folders from their own mailbox with other users via the User Portal (`/users/2/`), and those shares land in `user_folder_shares` rather than `shared_mailbox_permissions` . They are projected to `dovecot-acl` files by `sync_user_folder_acl_file.cfm` using the same vfile driver. The **Rebuild ACL Files** button on this page rebuilds both types of share in one pass, so admins don't have to think about the distinction when troubleshooting.

The two share types are otherwise independent:

	Admin-managed shared mailbox	User-initiated folder share
Surface	This page	User Portal > Folder Sharing
Storage	`shared_mailboxes` + `shared_mailbox_permissions`	`user_folder_shares`
Underlying mailbox	A dedicated `mailboxes` row with `mailbox_type='shared'`	The owner's existing mailbox + a named folder path
Visibility namespace	`Shared/<address>/INBOX`	`Shared/<owner>/<folder_path>`
ACL file path	`/srv/mail/<dom>/<local>/dovecot-acl`	`/srv/mail/<owner-dom>/<owner-local>/<folder>/dovecot-acl`
Cleanup on member removal	This page's Remove Permission	Owner removes the share from User Portal

Cross-domain members — not supported, enforced server-side

A shared mailbox on `company.com` can only be shared with users whose mailboxes are also on `company.com` . The same-domain rule is enforced in three places:

1. **Add Shared Mailbox modal** — the Initial Members list is filtered client-side to the selected domain.
2. **Manage Permissions modal** — the Tom Select picker is repopulated on open to only show users in the shared mailbox's domain.
3. **`add_permission` action handler** — compares `getUserMailbox.domain_id` against `getShared.domain_id` and returns error 26 on mismatch, so a forged form post can't bypass the UI filter.

The Dovecot shared namespace itself does not enforce this — the `acl_sharing_map` query keys on username, not domain — so the rule is a UX contract, not a Dovecot constraint. If you need a single inbox readable across multiple domains, the workable pattern is one shared mailbox per domain with a virtual recipient fan-out feeding both.

Nextcloud Mail caches the folder tree per account

Nextcloud Mail (the NC webmail app) caches each connected account's IMAP folder tree the first time the account is added and refreshes it lazily. **A user who is newly granted access to a shared mailbox via this page will NOT see it in Nextcloud Mail until they remove and re-add their NC mail account.** Standalone IMAP clients (Thunderbird, Outlook, Apple Mail) refresh the folder tree on the next IDLE cycle or manual sync, so they don't have this gotcha.

This is upstream NC Mail behavior, not a Hermes setting. The workaround is documented for end-users in the User Portal documentation; for admins, the remediation is to tell the affected user to re-add their NC mail account once the share is in place.

Feature-disabled behavior

When the Mailbox Sharing master switch on Settings is **off**:

- The Add / Rebuild / Manage Permissions buttons render disabled with a tooltip pointing back to Settings.
- An amber banner at the top of the page explains the state and links to Settings.
- Existing shared mailboxes appear in the table with status badge `Inactive (Sharing Off)` so the admin can see what would resume when the switch is flipped back on.
- The Delete button still works — admins can clean up rows while the feature is off.
- The `add_shared_mailbox`, `add_permission`, `edit_permission`, and `sync_all_acl_files` action handlers all check the master switch at entry and return error 31 if it's off, so a stale tab can't silently bypass the guard.

Dovecot itself does not declare the `Shared/` namespace when the master switch is off, so IMAP clients won't see shared folders even if the on-disk ACL files exist. Existing ACL files are preserved and re-activate as soon as the switch is flipped back on.

Failure semantics

What breaks	What happens
Master switch off + Add / Edit / Sync attempted	error 31, no DB write
Blank address prefix	error 10
Address prefix has invalid characters	error 11
Domain missing or not mailbox-type	error 12
Address collides with mailbox / alias / virtual recipient / existing shared mailbox	error 13
Quota not numeric or `<= 0`	error 14
Blank display name	error 15
Stale shared_mailbox_id (deleted between page load and submit)	error 21
Invalid user_mailbox_id	error 22
User already has permissions on this shared mailbox	error 23
Stale permission_id (Edit / Remove)	error 24
Add / Edit Permission with all seven flags off	error 25
Cross-domain member attempt	error 26
Any database operation throws inside the cftry	error 30, no rows committed
`doveadm mailbox create` fails	non-fatal — Maildir bootstraps via LMTP on first delivery instead
`sync_shared_mailbox_acl_file.cfm` fails	non-fatal — DB is the source of truth; the next permission change retries the sync, or admin can use Rebuild ACL Files
Maildir `rm -rf` on delete fails	non-fatal — DB rows are removed regardless; admin can manually clean up `/srv/mail/<domain>/<local>`

Files and containers touched

Path	Owner	Role
`config/hermes/var/www/html/admin/2/view_shared_mailboxes.cfm`	`hermes_commandbox`	Page + table + Add / Manage / Delete / Rebuild modals
`config/hermes/var/www/html/admin/2/inc/shared_mailbox_actions.cfm`	`hermes_commandbox`	Dispatcher for all six actions (add / delete / add_permission / edit_permission / remove_permission / sync_all_acl_files)
`config/hermes/var/www/html/admin/2/inc/sync_shared_mailbox_acl_file.cfm`	`hermes_commandbox`	Rebuilds one `dovecot-acl` file from `shared_mailbox_permissions`

Path	Owner	Role
`config/hermes/var/www/html/admin/2/inc/sync_user_folder_acl_file.cfm`	`hermes_commandbox`	Same engine for user-initiated folder shares
`config/hermes/var/www/html/admin/2/inc/get_shared_mailbox_permissions_json.cfm`	`hermes_commandbox`	AJAX endpoint for the Manage Permissions table
`/srv/mail/<domain>/<local>/dovecot-acl`	`hermes_dovecot` (vmail:vmail 0660)	Per-mailbox vfile ACL file — Dovecot 2.4's enforcement source
`/srv/mail/<domain>/<local>/`	`hermes_dovecot`	The Maildir itself
`/opt/hermes/tmp/<token>_sync_shared_acl.sh`	`hermes_commandbox`	Throwaway shell script used to ship the ACL payload through `docker exec -i` via heredoc
`shared_mailboxes`, `shared_mailbox_permissions`, `user_folder_shares`, `mailboxes`, `recipients`, `maddr`, `sender_login_maps`, `dovecot_acl`, `dovecot_acl_shared`, `parameters2`	`hermes_db_server`	Storage
`hermes_dovecot` container	—	`doveadm mailbox create` (bootstrap), `rm -rf` (delete), and the in-container `mkdir / cat / chown / chmod` invoked by the sync helper

Related

- Settings — the Mailbox Sharing master switch. Must be on for shared mailboxes to actually function at the IMAP layer. Also the Dovecot TLS profile and connection limits that all shared-mailbox access goes through.
- Mailboxes — the user mailbox list. Members granted permission on this page must already exist there.
- Domains — the mailbox domain list. A shared mailbox is anchored to exactly one domain; cross-domain sharing is not supported.
- Aliases — if you want one inbound address to deliver into one mailbox (rather than be visible to several users), an alias is the lighter-weight option. Aliases have no ACL surface at all.
- Email Relay > Virtual Recipients — the relay-side fan-out pattern. Sometimes a virtual recipient feeding two shared mailboxes (one per domain) is the right tool when a single role address needs to be visible to users on more than one mailbox domain.
- Mailbox Rules — Sieve rules can be configured on shared mailboxes the same way as on user mailboxes; the authentication path is the granting user, not the shared address.
- Authentication Settings — Submission-port auth that the Send-As flag piggybacks on, plus the LDAP backend that Dovecot looks up members against.

Revision #14
Created 2026-05-31 12:52:18 UTC by Dino Edwards
Updated 2026-06-13 12:30:15 UTC by Dino Edwards