

# Domains

# Domains

Admin path: **Email Server > Domains** (`view_mailbox_domains.cfm`, `inc/mailbox_domain_add_action.cfm`, `inc/mailbox_domain_edit_action.cfm`, `inc/mailbox_domain_delete_action.cfm`, `inc/get_mailbox_domain_json.cfm`, `inc/sync_mailbox_sans.cfm`, `inc/generate_nginx_configuration.cfm`, `inc/generate_transports.cfm`, `inc/generate_relay_domains.cfm`, `inc/generate_postfix_configuration.cfm`, `inc/add_domain_djigzo.cfm`, `inc/delete_domain_djigzo.cfm`).

This page manages the list of **mail-server domains** — the SMTP domains for which Hermes is itself the destination MTA, accepting inbound mail via Postfix and delivering it locally over LMTP to Dovecot mailboxes on `/mnt/vmail`. Each row pairs a `domains` row (`type='mailbox'`) with a `mailbox_domains` row (the per-domain SAN certificate binding) plus a `transport` row hardwired to `lmtp:[hermes_dovecot]:24`, a `senders` row, and a domain-wide `recipients` row carrying the default Amavis SVF policy.

This is the **mailbox-topology** counterpart to [Email Relay > Domains](#). Both pages edit the same `domains` table but use the `type` column to partition rows: `type='relay'` belongs to the Relay page and forwards mail downstream; `type='mailbox'` belongs to this page and delivers mail locally. A single installation can run any mix of the two topologies — see [Email Relay > Domains § Hermes topology overview](#) for the high-level diagram.

“ **Not to be confused with [Email Relay > Domains](#)**. The Relay page handles domains where Hermes forwards mail to a downstream MX (M365, Exchange, Google Workspace, an internal hub). This page handles domains where Hermes IS the final destination — mailboxes, IMAP/POP3, Submission, ManageSieve, Nextcloud Mail, autodiscover/autoconfig, DAV — backed by Dovecot.

## Configuration storage

A single Add Mailbox Domain submission writes (or upserts) **five** rows across four tables and regenerates Postfix + Nginx + Ciphermail:

Table	Role
<code>domains</code>	One row per mailbox domain. <code>type='mailbox'</code> partitions it from the Relay page. Mailbox-specific metadata lives here: <code>default_quota_mb</code> (default per-mailbox quota in MB), <code>catchall_mailbox</code> (optional <code>postmaster@domain</code> style address), <code>nextcloud_enabled</code> (per-domain default — controls whether new mailboxes get a Nextcloud account), <code>enforce_mfa</code> (per-domain default for 2FA), <code>org_name</code> / <code>org_phone</code> / <code>org_address</code> / <code>org_website</code> / <code>org_logo_path</code> (Pro Organization Information for signature placeholder substitution), <code>allow_user_signatures</code> (gates the user-portal personal-signature editor for this domain).
<code>mailbox_domains</code>	One row per mailbox domain. <code>mailbox_certificate</code> foreign-keys into <code>system_certificates</code> — the per-domain TLS cert used by Dovecot IMAP/POP3/Submission, the autodiscover/autoconfig vhosts, and the DAV per-domain vhost.
<code>mailbox_sans</code>	One row per SAN prefix × domain (built from <code>additional_sans</code> ). Drives per-SAN DNS/IP probe state for the certificate validator.
<code>transport</code>	Always <code>lmtp:[hermes_dovecot]:24</code> — mail-server domains never use SMTP forwarding.
<code>senders</code> + <code>recipients</code>	<code>senders.sender = domain</code> , <code>recipients.recipient = @domain</code> with <code>domain='1'</code> + the default <code>spam_policies</code> policy attached so Amavis runs on every inbound message.

The mailbox-domain row in `domains` deliberately reuses many columns from the relay path so the Postfix generators (`generate_transports`, `generate_relay_domains`, `generate_postfix_configuration`) treat both topologies uniformly — the only thing that differs is the transport string and the per-mailbox personal info / org info columns.

## How a mailbox domain becomes live config

```
form submit → mailbox_domain_add_action.cfm
|
| validate domain + cert mode (Pro gate on 'auto')
| duplicate-check against domains.domain
|
```

```

| --- write DB ---
| INSERT transport (lmtpl:[hermes_dovecot]:24)
| INSERT senders (sender = domain, action = OK)
| INSERT recipients(recipient = @domain,
|                 domain='1', policy_id=default,
|                 status='OK')
| INSERT domains (... , type='mailbox', default_quota_mb,
|                 catchall_mailbox, nextcloud_enabled,
|                 enforce_mfa, created_at, updated_at)
| UPSERT mailbox_domains (domain, mailbox_certificate)
|
| --- regenerate ---
v
sync_mailbox_sans.cfm          -> mailbox_sans (one per prefix)
generate_transports.cfm       -> /etc/postfix/transport + postmap
generate_relay_domains.cfm    -> /etc/postfix/relay_domains
generate_postfix_configuration.cfm
                               -> /etc/postfix/main.cf
                               + postfix reload (docker exec)
generate_nginx_configuration.cfm
                               -> per-domain Nginx vhosts
                               (autodiscover, autoconfig, DAV)
add_domain_djigzo.cfm         -> registers domain in CIPHERMAIL
occ group:add <domain>       -> Nextcloud group (if NC enabled)
                               (docker exec hermes_nextcloud)
|
v
preload_restart_nginx.cfm?returnUrl=... (Nginx restart, then redirect)

```

Edit follows the same shape minus the inserts (UPDATE on `domains`, UPSERT on `mailbox_domains`, re-sync SANs, regen Nginx). Delete reverses the writes after running dependency checks (see Delete below).

## Fields on the page

### Add Mailbox Domain card

Field	Default	Notes
<b>Domain Name</b>	(empty)	Trimmed, lower-cased, validated by the email-trick. Rejected if the domain already exists in <code>domains</code> (as relay or mailbox). The <code>mailbox_domains</code> table is allowed to have a pre-existing row (left over from prior ACME work) — it gets UPSERTed in place.
<b>Default Quota (GB)</b>	5	Per-domain default for new mailboxes. Stored in DB as MB ( <code>default_quota_mb</code> ). 0.5 GB minimum, 1024 GB max, 0.5 GB step. The per-mailbox quota is set on <a href="#">Mailboxes</a> ; this is the value pre-filled when adding a new mailbox under the domain.
<b>Catch-All Mailbox</b>	(empty)	Optional. An existing mailbox address that receives mail for any unknown recipient at the domain. Free-text — admin's responsibility to point at a real mailbox.
<b>SAN Certificate — Auto-managed (Let's Encrypt)</b>	Pro: checked / Community: disabled	<i>Pro Edition only.</i> Creates a placeholder Acme row in <code>system_certificates</code> ; the certificate validator then validates SAN DNS + IP, requests the cert, and auto-renews. Zero maintenance once DNS is in place.
<b>SAN Certificate — Use existing certificate</b>	Community: checked	Pulls from <code>system_certificates</code> where <code>san='1'</code> OR the row is a system-flagged placeholder. The dropdown labels system placeholders as <code>TEMPORARY PLACEHOLDER (replace before production)</code> and sorts them last so the default is a real SAN cert.
<b>Enable Nextcloud webmail for this domain</b>	unchecked	Per-domain default for new mailboxes. When checked, creates a Nextcloud group named after the domain (via <code>occ group:add</code> ) and pre-fills the Nextcloud toggle on the <a href="#">Add Mailbox</a> form. Does <b>not</b> retroactively enable NC for existing mailboxes.
<b>Require Two-Factor Authentication for this domain</b>	unchecked	Per-domain default for new mailboxes. Same convention as Nextcloud — defaults only, no cascade to existing rows.

## Mailbox domains table

Sortable, searchable, exportable. Columns:

Column	Source	Badge logic
Domain	<code>domains.domain</code>	Plain text
Certificate	<code>system_certificates.friendly_name</code> via <code>mailbox_domains.mailbox_certificate</code>	Link to <code>view_system_certificates.cfm</code> ; badge <code>Auto (LE)</code> for <code>type='Acme'</code> , <code>Imported</code> otherwise; <code>Missing</code> if no binding
Cert Status	derived from <code>mailbox_sans</code> rows for the domain	<code>Verified</code> (all SANs DNS-confirmed) / <code>Partial</code> / <code>Awaiting Cert</code> / <code>Pending</code> / <code>DNS Failed</code> / <code>No SANs</code> / <code>No Cert</code> . Imported certs always show <code>Imported</code> .
Default Quota	<code>default_quota_mb</code>	Rendered in GB
Catch-All	<code>catchall_mailbox</code>	Em-dash if NULL
Nextcloud	<code>nextcloud_enabled</code>	<code>Enabled</code> (success) / <code>Disabled</code> (secondary)
2FA	<code>enforce_mfa</code>	<code>Required</code> (success) / <code>Optional</code> (secondary)
DKIM	aggregated from <code>dkim_sign</code>	<code>Active</code> / <code>Disabled</code> / <code>None</code> — same logic as the Relay page
Actions	—	Edit (opens modal), DNS Records (opens helper modal), DKIM Keys (→ <code>edit_domain_dkim.cfm</code> ), Delete

## Edit Mailbox Domain modal

Opens via `openEditModal(id)`, fetches `./inc/get_mailbox_domain_json.cfm` over AJAX, hydrates every form field. **Domain Name is read-only on edit** — same convention as the Relay page (renaming a domain across all the joined tables is risky enough that the page enforces add-and-delete instead).

The Edit modal carries everything from Add plus three extra sections that exist only after creation:

Section	Notes
<b>Organization Information</b> ( <i>Pro only</i> )	<code>org_name</code> , <code>org_phone</code> , <code>org_address</code> , <code>org_website</code> . Used by the body milter's signature substitution to fill <code>{{org.name}}</code> , <code>{{org.phone}}</code> , <code>{{org.address}}</code> , <code>{{org.website}}</code> placeholders in organizational signatures. See <a href="#">Organizational Signatures</a> . All fields optional. Community installs see a Pro upsell badge and the inputs are HTML-disabled — the action handler also skips the UPDATE on Community so a tampered form post can't write data and existing values survive a Pro→Community downgrade.

Section	Notes
<code>org_logo_path</code>	Column exists but no UI yet — placeholder for follow-up integration with the inline image pipeline that ships organizational signature logos.
<b>Allow users in this domain to manage their own signatures</b>	Per-domain toggle ( <code>allow_user_signatures</code> , both tiers). When on, mailbox users see a Signature page in <code>/users/2/</code> . When off, the page is hidden and any user-edited signature rows for the domain are ignored at send time. The body milter respects this on the next signature-map regen.

The modal explicitly tags `Nextcloud webmail` and `Two-Factor Authentication` as **defaults for new mailboxes** — toggling them does **not** flip the corresponding per-mailbox flags on existing rows. To change an existing mailbox use the per-mailbox Edit Options dialog on [Mailboxes](#).

## DNS Records modal

Per-domain reference card surfacing every DNS record an operator needs to publish for the domain to actually receive mail and support client auto-discovery: MX, autoconfig/autodiscover CNAMEs, the SRV chain (`_imap`, `_imaps`, `_pop3`, `_pop3s`, `_submission`, `_submissions`, `_sieve`, `_autodiscover`), CalDAV/CardDAV SRV+TXT (`_caldavs`, `_carddavs` with `path=/nc/remote.php/dav/`), plus example SPF and DMARC TXT records. DKIM TXT records are listed separately under DKIM Keys.

Console host (`parameters2 console.host`) is interpolated into every record so the values are copy-paste ready.

## Delete Mailbox Domain modal

Confirms the destructive action. The handler runs two dependency checks before allowing the delete:

Check	If it returns rows →
Mailboxes under this domain ( <code>mailboxes.domain_id = &lt;id&gt;</code> )	Error 16, abort, link admin to <a href="#">Mailboxes</a> to clear them first
Recipients still attached to the domain (excluding the domain-wide <code>@domain</code> row)	Error 17, abort

If both pass, the handler:

1. Captures the bound `mailbox_certificate` id (for orphan-cert detection).
2. Deletes `mailbox_domains`, `domains`, `transport`, `senders`, `recipients` (the five rows linked at creation).
3. Deletes the domain's `mailbox_sans` rows **directly** (does not call `sync_mailbox_sans.cfm` — sync would nuke validated IP/DNS state on other domains if it ran during a delete→re-add

cycle).

4. Regenerates Postfix + Nginx, deregisters from CIPHERMAIL, runs `occ group:delete <domain>` against Nextcloud (non-fatal).
5. If the bound certificate now belongs to no other mailbox domain, surfaces an **Orphaned Certificate** flash on the next page render pointing the admin to [System Certificates](#). The cert is **not** auto-deleted because Let's Encrypt limits duplicate certificate issuance to 5 per week and accidentally throwing away a cert you might re-need is a non-recoverable mistake.

“ **Operational consequence — mailbox data on disk is NOT deleted.** The delete handler removes the Dovecot domain wiring (transport, recipient acceptance, cert binding) but does **not** touch `/mnt/vmail/<domain>/`. If you intend to permanently retire a domain, remove the mailbox directories from the host after the delete completes.

## Per-domain Nginx vhosts

Each mailbox domain generates per-domain Nginx vhosts for:

- `autodiscover.<domain>` — Outlook / iOS Mail auto-configuration
- `autoconfig.<domain>` — Thunderbird / K-9 Mail auto-configuration
- The DAV chain via the SRV records published by the DNS Records modal

Add and Edit both call `generate_nginx_configuration.cfm` then redirect through `preload_restart_nginx.cfm` (the canonical restart pattern that avoids the brief `ERR_CONNECTION_REFUSED` blip in user-driven flows).

“ **Known gotcha — editing the vhost template does NOT update already-generated vhosts.** The generator writes per-domain files at install time and on subsequent saves. If the underlying template (in `/opt/hermes/templates/`) is hand-edited, existing vhost files stay stale until each domain is re-saved (or until a separate re-render pass is run). Operators changing the template should plan for a bulk re-save afterwards.

## Cert SAN binding and the validator

`sync_mailbox_sans.cfm` reads `additional_sans` (the global list of prefixes — `mail.`, `autodiscover.`, `autoconfig.`, plus any custom ones) and writes one `mailbox_sans` row per prefix × this domain, pointing at the selected certificate. Each row carries IP and DNS probe state.

A separate scheduled task (System > [SAN Management](#)) walks `mailbox_sans` every 30 minutes, probes each subdomain for the expected IP and DNS A/CNAME record, and updates `ip_result_msg` / `dns_result_msg`. The Cert Status column on the main table summarizes these results.

For Pro Edition's auto-managed certs the validator then triggers a Let's Encrypt issuance once every SAN passes both probes. For imported certs the probes are informational only — the cert is trusted as-is.

See [SAN Management](#) for the full SAN editor.

## Failure semantics

What breaks	What happens
Domain name empty	<code>session.m = 10</code> , redirect, no DB write
Domain name fails email-trick validation	<code>session.m = 11</code> , redirect, no DB write
Domain already exists in <code>domains</code> (relay or mailbox)	<code>session.m = 12</code> , redirect, no DB write
Auto-managed selected on Community edition	<code>session.m = 14</code> , redirect, no DB write
<code>cert_id</code> invalid for <code>Use existing</code>	<code>session.m = 13</code> , redirect, no DB write
<code>default_quota_gb</code> not a positive number	<code>session.m = 15</code> , redirect, no DB write
Delete blocked: mailboxes still exist	<code>session.m = 16</code> , redirect, abort. Detail count shown in the alert.
Delete blocked: recipients still exist	<code>session.m = 17</code> , redirect, abort
<code>add_domain_djigzo.cfm</code> errors during Ciphermail registration	Domain is already in the DB; encryption gateway will not know about the domain until the next re-save. Non-fatal.
<code>occ_group:add</code> fails (NC down, group exists)	Non-fatal <code>cftry</code> — mailbox-domain creation still succeeds; admin can re-toggle in Edit to retry
Nginx vhost regen fails	Domain is in the DB; per-domain auto-discovery URLs will return errors until the next successful Edit/regen
Postfix reload fails	Live config keeps the previous values; reload error is in container logs

## Files and containers touched

Path	Owner	Role
<code>config/hermes/var/www/html/admin/2/view_mailbox_domains.cfm</code>	hermes_commandbox	Page + Add card + Edit/Delete/DNS modals
<code>config/hermes/var/www/html/admin/2/inc/mailbox_domain_add_action.cfm</code>	hermes_commandbox	Add handler
<code>config/hermes/var/www/html/admin/2/inc/mailbox_domain_edit_action.cfm</code>	hermes_commandbox	Edit handler
<code>config/hermes/var/www/html/admin/2/inc/mailbox_domain_delete_action.cfm</code>	hermes_commandbox	Delete handler
<code>config/hermes/var/www/html/admin/2/inc/get_mailbox_domain_json.cfm</code>	hermes_commandbox	AJAX hydrator for the Edit modal
<code>config/hermes/var/www/html/admin/2/inc/sync_mailbox_sans.cfm</code>	hermes_commandbox	Builds <code>mailbox_sans</code> rows from <code>additional_sans</code> × domain
<code>config/hermes/var/www/html/admin/2/inc/generate_nginx_configuration.cfm</code>	hermes_commandbox	Per-domain vhost generator
<code>config/hermes/var/www/html/admin/2/inc/generate_transports.cfm</code> / <code>generate_relay_domains.cfm</code> / <code>generate_postfix_configuration.cfm</code>	hermes_commandbox	Shared Postfix regenerators (also used by <a href="#">Email Relay &gt; Domains</a> )
<code>config/hermes/var/www/html/admin/2/inc/add_domain_djigzo.cfm</code> / <code>delete_domain_djigzo.cfm</code>	hermes_commandbox	Ciphermail registration
<code>config/hermes/var/www/html/admin/2/inc/signature_regen_map.cfm</code>	hermes_commandbox	Rebuilds the body milter's <code>signature_by_sender</code> map + <code>sender_data.json</code> after org info / <code>allow_user_signatures</code> edits
<code>config/hermes/var/www/html/admin/2/preload_restart_nginx.cfm</code>	hermes_commandbox	Nginx restart shim used on Add and Edit redirect
<code>/etc/postfix/transport + .db,</code> <code>/etc/postfix/relay_domains,</code> <code>/etc/postfix/main.cf</code>	hermes_postfix_dkim	Postfix maps regenerated on every save
Per-domain Nginx vhost files	hermes_nginx (mounted)	Generated by <code>generate_nginx_configuration.cfm</code>
<code>domains,</code> <code>mailbox_domains,</code> <code>mailbox_sans,</code> <code>transport,</code> <code>senders,</code> <code>recipients</code>	hermes_db_server	The mailbox-domain row group
<code>system_certificates,</code> <code>additional_sans</code>	hermes_db_server	Cert inventory + SAN prefix list
hermes_nextcloud container	—	<code>occ group:add / group:delete &lt;domain&gt;</code> for the per-domain NC group
hermes_ciphermail container	—	Domain registration via CLITool

Every shell-out uses `docker exec ...` per the standard Hermes pattern.

# Related

- [Email Relay > Domains](#) — the relay topology twin. Mailbox and relay domains share the same `domains` table but partition on `type`. **Do not confuse with this page.**
- [Email Server > Mailboxes](#) — per-mailbox CRUD. A mailbox domain is meaningless without mailboxes; add the domain here first, then add mailboxes there.
- [Email Server > Settings](#) — global Dovecot configuration (TLS profile, compression, encryption at rest, quota warning thresholds). The per-domain default quota set here is what Email Server > Settings's warning thresholds measure against on a per-mailbox basis.
- [Email Server > Aliases](#) — alias addresses that resolve to local mailboxes within a mailbox domain.
- [Email Server > Shared Mailboxes](#) — shared mailboxes are per-domain just like regular mailboxes.
- [Email Server > Mailbox Rules](#) — per-mailbox Sieve rules.
- [Email Server > SAN Management](#) — the global SAN prefix list (`additional_sans`) that `sync_mailbox_sans.cfm` multiplies against every mailbox domain.
- [System Certificates](#) — certificate inventory that the SAN Certificate dropdown draws from, including the bootstrap placeholder cert.
- [LDAP RemoteAuth](#) — mailbox users can authenticate against an upstream LDAP/AD using the same `auth_type='remote'` pattern documented for relay recipients.
- [Organizational Signatures](#) (*Pro*) — consumer of the Organization Information fields on the Edit modal.

---

Revision #8

Created 2026-05-31 12:52:14 UTC by Dino Edwards

Updated 2026-05-31 14:01:14 UTC by Dino Edwards