Relay Recipients

Relay Recipients

Admin path: **Email Relay > Relay Recipients** (`view_internal_recipients.cfm`, `add_internal_recipients.cfm`, `edit_internal_recipient_backend.cfm`, `inc/delete_internal_recipients.cfm`, `inc/edit_internal_recipients.cfm`, `inc/edit_internal_recipients_djigzo.cfm`, `inc/get_int_recipient_json.cfm`, `inc/send_recipient_welcome_email.cfm`, `inc/send_recipient_welcome_email_remoteauth.cfm`).

> **The page filename is `view_internal_recipients.cfm`, not `view_relay_recipients.cfm`.** The original concept was "internal" recipients (mail accepted into the gateway and forwarded to an internal backend); the UI label was renamed to **Relay Recipients** in commit `c547fdd9` but the filename, table column `recipients.recipient_type='relay'`, and several handler names still carry the legacy `internal_recipients` naming. Treat the two terms as synonymous.

This page manages the **per-address recipient roster** for relay-mode domains — the list of mailboxes Hermes accepts inbound mail for and forwards downstream, and the list of authenticated senders that can relay outbound mail through the gateway. Each row in the `recipients` table is one email address with a stack of per-recipient settings: SVF policy, quarantine notifications, encryption flags (PDF/S/MIME/PGP), S/MIME certificate + PGP keyring slots, backend override, auth mode (local vs RemoteAuth), and 2FA enforcement.

This is the **recipient-validation** half of the relay topology. Pairs with Domains (the domains those recipients live under), Relay Networks (the trusted source IPs), and Virtual Recipients (alias-only addresses that forward without a real account).

Relay Recipient vs Virtual Recipient vs Mailbox

Three different recipient concepts share the email-address namespace in Hermes — keep them straight:

Concept	Stored in	Has a local account?	Delivered to
Relay Recipient (this page)	`recipients` where `recipient_type='relay'`, `domain IS NULL`	Yes — LDAP entry + optional app passwords	Downstream MX (per `domains` row's `transport`)
Virtual Recipient	`virtual_recipients`	No — alias only	Rewrites to another address, which then needs a Relay Recipient or external destination
Mailbox	`mailboxes` (separate `mailbox_domains` topology)	Yes — Dovecot mailbox	Local Dovecot LMTP at `/mnt/vmail`

A Relay Recipient is the only one of the three that authenticates for outbound submission (SMTP AUTH on port 587) and for web/portal login (via Authelia). Virtual Recipients are pure forwarding rules; Mailboxes are the mail-server-topology equivalent. See Email Server > Mailboxes for the Mailbox flow.

What a Relay Recipient row carries

```
recipients table  (one row per email address)
├── recipient                jsmith@company.com
├── recipient_type           'relay'
├── domain                   NULL   (domain rows use domain='1')
├── auth_type                'local' | 'remote'
├── remoteauth_domain        NULL if local; mapping key if remote
├── enforce_mfa              0 | 1   (admin policy — see #225 Phase 2)
├── policy_id  ──────────►   spam_policies.policy_id (SVF policy)
├── pdf_enabled / smime_enabled / pgp_enabled / digital_sign
├── backend_server / backend_port / backend_tls   (per-recipient override)
└── (cert+keyring slots populated lazily by the queue)
```

Side tables linked at create/edit time:

Table	What it stores
`user_settings`	Per-user portal toggles (`report_enabled`, `train_bayes`, `download_msg`), `ldap_username`, mailbox flags
`recipient_certificates`	S/MIME certs issued for the recipient (lazy — populated by `cert_generation_queue`)

Table	What it stores
`recipient_keystores`	PGP keyrings (lazy — same queue)
`app_passwords`	Per-application passwords (Argon2-hashed) for IMAP/SMTP/CalDAV/CardDAV/Nextcloud — see Credential Model
`wblist`	Whitelist/blacklist entries owned by the recipient
`cert_generation_queue`	Pending S/MIME and PGP generation jobs

Add Recipient(s) — `add_internal_recipients.cfm`

The Add Recipient(s) button navigates to a multi-line input form that creates many recipients in one submission. Three add modes:

Local-auth bulk add — one email per line

When **Auth Type** is `Local` (the default), the textarea takes one email per line. The page generates a random password for each new recipient, sends a welcome email via `send_recipient_welcome_email.cfm` that includes a **first-login password-reset link**, and stores the LDAP entry with a placeholder `userPassword` that will be overwritten when the user follows the link.

```
jsmith@company.com
jdoe@company.com
bob.smith@company.com
```

RemoteAuth bulk add — same line format

When **Auth Type** is `Remote` and the selected mapping's DN pattern only uses `{username}` and/or `{email}`, the textarea is still one email per line. No password is generated — the recipient authenticates against the upstream LDAP/AD via the `remoteauth` overlay (see LDAP RemoteAuth). The welcome email goes through `send_recipient_welcome_email_remoteauth.cfm` and tells the user to sign in with their **organization password**, not a Hermes-issued one.

RemoteAuth CSV add — `First,Last,Email` per line

When the RemoteAuth mapping's DN pattern uses `{firstname}` or `{lastname}` (typical for AD `cn=` patterns), the textarea **switches to CSV mode** because email-only input doesn't carry enough data to expand the pattern. Header rows (`"GivenName","Surname","Mail"`) are auto-detected and skipped, and unknown columns are ignored.

Source	Command / file shape
PowerShell	`Get-ADUser -Filter * -Properties GivenName,Surname,Mail | Select GivenName,Surname,Mail | Export-Csv users.csv -NoTypeInformation`
CSVDE (Windows Server built-in)	`csvde -f users.csv -l "givenName,sn,mail"`
Excel / manual	Three columns saved as CSV

See LDAP RemoteAuth § Adding RemoteAuth users in bulk for the full CSV format reference.

The Add form also accepts the same per-recipient stack of options as the Edit Options modal (SVF policy, quarantine notifications, etc.) — those defaults are written to every new row in one shot.

The Recipients table

Sortable, searchable, exportable (copy/CSV/Excel/PDF/print via DataTables Buttons; `stateSave: true`). Columns:

Column	Source	Notes
Checkbox	—	Multi-select for the action buttons above the table
S/MIME	link to `view_recipient_certificates.cfm?type=1&id=…`	Per-recipient cert manager
PGP	link to `view_recipient_keyrings.cfm?type=1&id=…`	Per-recipient keyring manager
Recipient	`recipients.recipient`	Email address
Auth	`recipients.auth_type` + `remoteauth_domain`	`LOCAL` badge (secondary) or `REMOTE` badge (primary, tooltip shows mapping key)
Backend	`recipients.backend_server[:port]`	Per-recipient override or `(domain default)` placeholder

Column	Source	Notes
2FA	LDAP `cn=two_factor` + `enforce_mfa`	**Two independent pills** — see Two-pill 2FA column below
Policy	`policy.policy_name` via join	Assigned SVF policy
Quarantine Notifications	`user_settings.report_enabled`	`YES` / `NO` badge
Train Bayes	`user_settings.train_bayes`	`YES` / `NO`
Download Msgs	`user_settings.download_msg`	`YES` / `NO`
PDF / S/MIME / PGP Encrypt	per-row encryption flags	`YES` / `NO` badges
Sign All	`recipients.digital_sign`	`YES` / `NO`
S/MIME Cert	join against `recipient_certificates`	`YES` (green badge) if a cert exists
PGP Keyring	join against `recipient_keystores`	`YES` (green badge) if a keyring exists

The query filters `WHERE recipients.domain IS NULL AND (recipient_type = 'relay' OR recipient_type IS NULL)` so only relay-mode rows appear — mailbox-topology rows (with `recipient_type='mailbox'`) are managed under Email Server > Mailboxes.

Two-pill 2FA column

The 2FA column shows **two orthogonal states** as independent pills, because admin enforcement and user enrollment are decoupled (#225 Phase 1.5 + Phase 2):

Pill	Source	Means
Enrolled (success badge)	LDAP `cn=two_factor` group membership	The user has registered a 2FA device (TOTP, security key, or Duo Push) and Authelia challenges them at sign-in
Required (warning badge)	`recipients.enforce_mfa = 1`	Admin policy demands 2FA. The recipient sees an urgent banner in the user portal directing them to Account Settings until they enroll

Enrolled	Required	What it looks like	Means
no	no	em-dash	Default state. No 2FA.
yes	no	Enrolled only	Voluntary enrollment. User opted in; admin doesn't enforce.
no	yes	Required only	Admin set the policy; user hasn't yet registered a device.

Enrolled	Required	What it looks like	Means
yes	yes	Both pills	Required and complied with.

The single LDAP `ldapsearch` query against `cn=two_factor,ou=groups,dc=hermes,dc=local` runs once per page render, then each row checks for its DN substring in the result — avoids N+1 LDAP roundtrips.

Bulk action buttons

Button	Action	Selection requirement
Create Recipient(s)	Navigates to `add_internal_recipients.cfm`	—
Edit Options	Opens the Edit Options modal	At least one row
Edit Encryption	Opens the Edit Encryption modal	At least one row
Edit Backend	Navigates to `edit_internal_recipient_backend.cfm?ids=…`	At least one row
Reset 2FA Devices	Opens the Reset 2FA Devices modal	At least one row
Delete	Opens the delete-confirm modal	At least one row

Selecting zero rows and clicking any of the edit/delete buttons surfaces an alert (`Please select at least one recipient`) instead of opening the modal.

Edit Options modal — AJAX pre-fill vs bulk-edit warning

The Edit Options modal handles SVF policy, quarantine notifications, Train Bayes, Download Messages, and 2FA enforcement (`enforce_mfa`). It has **two modes**, selected by the JS based on how many rows are checked:

Single-select: AJAX pre-fill

When exactly one row is checked, the JS calls `./inc/get_int_recipient_json.cfm?id=<rid>` over POST and hydrates every form field with that recipient's current values before opening the modal. The admin sees the recipient's actual policy, current notification mode, current `enforce_mfa` state, etc.

— submit edits only what changed.

Multi-select: bulk-edit warning

When 2+ rows are checked, the modal shows a prominent red **Bulk edit — N recipients selected** alert at the top:

> The fields below are **not pre-filled from each recipient's current settings** — they show the form's default values. Submitting will **OVERWRITE every field on every selected recipient** with whatever you see now.

The 2FA-specific footnote then warns that leaving the Two-Factor Authentication dropdown at `Disable` will reset every selected recipient's `enforce_mfa` to `0` — but **the user is not removed from `cn=two_factor` automatically** (the LDAP cascade only fires on 0→1 transitions). To strip an existing enrollment, the admin must use the Reset 2FA Devices modal with the nuclear-option checkbox.

This is intentional — the bulk-edit form has been a foot-gun in the past (admins thinking "Disable" only changed the one row), so the warning is unmissable. The recommended pattern: **edit a single recipient with their current values pre-filled, select only one row.**

Edit Encryption modal

Handles `pdf_enabled`, `smime_enabled`, `digital_sign`, `pgp_enabled`, and the cert/keyring generation parameters (CA, validity, key size, algorithm, PGP key length). Submit triggers `edit_internal_recipients_djigzo.cfm` which updates the row and **queues async S/MIME cert + PGP keyring generation** into `cert_generation_queue` if the flags flip on and no existing cert/keyring is present.

The page renders a **Background Generation in Progress** info banner while `cert_generation_queue` has any `pending` or `processing` rows, and a **Generation Failures** warning with a **Retry Failed Jobs** button if any rows are in `failed` state. The Retry button updates matching rows to `status='pending', error_message=NULL, started_at=NULL` so the next scheduler tick re-attempts them.

Edit Backend page

Per-recipient override of the downstream backend server / port / TLS mode. The default is `NULL` on all three columns, which falls back to the parent domain's `transport` row (set on the Domains page). Useful for routing specific recipients to a different MX — e.g., a single user whose mailbox is on a different server than the rest of the domain.

The Backend column on the main table shows the override host (and port via tooltip) or `(domain default)` for the fallback case.

Reset 2FA Devices modal

Replaces the older "Recipient Access Control" modal as of #225 Phase 2. The one_factor/two_factor radio is gone — the canonical admin policy is the **Two-Factor Authentication** select on Edit Options. This modal is now single-purpose: clear Authelia TOTP/WebAuthn devices for the selected recipients via `docker exec hermes_authelia authelia storage user totp/webauthn delete`.

Two modes:

Mode	What it does
Default	Deletes TOTP + WebAuthn device registrations in Authelia. User stays under 2FA enforcement and re-registers on next sign-in. "User lost their phone" recovery.
Nuclear (checkbox)	Also moves the user from `cn=two_factor` back to `cn=one_factor`. Admin override of voluntary enrollment, or full account reset.

> **Does not affect Duo Push.** Duo enrollments live on Duo's cloud servers, not in Authelia's database. Use the Duo Admin Console for Duo device management.

> **Cascade interaction.** If the per-recipient `enforce_mfa` policy in Edit Options is still `Enable`, the nuclear option's removal from `cn=two_factor` will be **reversed** on the next save of the Edit Options modal (the 0→1 LDAP cascade fires again). To truly de-enforce, set `enforce_mfa = Disable` first.

Delete

The Delete modal confirms the irreversible action. The `delete_internal_recipients.cfm` handler then runs an unusually-long cleanup sequence per recipient — the kind of cascade that makes orphan

rows the rule when CFML deletes are skimped:

```
For each selected recipient ID:
1. Look up ldap_username via user_settings join
2. docker exec hermes_authelia authelia storage user totp delete <user>
3. docker exec hermes_authelia authelia storage user webauthn delete <user> --all
4. ldap_delete_user_relay.cfm — remove LDAP stub entry + group memberships
5. Cancel any pending password_reset_requests rows for this email
6. DELETE FROM recipients WHERE id = <rid>
7. DELETE FROM recipients_temp WHERE recipient = <email>
8. DELETE FROM wblist WHERE rid = <rid>
9. DELETE FROM user_settings WHERE email = <email>
10. DELETE FROM mailaddr (and wblist by sid) for the address
11. Delete recipient_certificates + cm_keystore from djigzo
12. (caller continues with the next ID)
```

Steps 2–3 prevent a re-created recipient at the same email from silently inheriting the prior owner's TOTP/WebAuthn enrollments. Failures inside `cftry` blocks are non-fatal — the desired end-state ("no devices") is achieved whether or not the user had anything enrolled in the first place.

> **Known gap (#102).** When a Relay Recipient with `auth_type='remote'` is deleted, the deletion of the LDAP stub entry happens, but the RemoteAuth domain-mapping deletion validation in `view_remoteauth.cfm` / `edit_remoteauth_mapping.cfm` does **not** check the `mailboxes` table yet (it only checks `system_users` and `recipients`). When RemoteAuth is wired to mailboxes, that validation must add a third query. Not a bug today — relay recipients are correctly covered — but a forward-looking integration point. See LDAP RemoteAuth § Deletion validation.

Local-auth vs RemoteAuth — the credential split

Aspect	`auth_type = 'local'`	`auth_type = 'remote'`
Web portal sign-in	Hermes LDAP `userPassword` (user sets via reset link)	Upstream AD/LDAP via overlay; Hermes never sees the password

Aspect	`auth_type = 'local'`	`auth_type = 'remote'`
IMAP / SMTP / CalDAV / CardDAV / NC	`app_passwords` row (Argon2-hashed in Hermes DB)	Same — `app_passwords` row in Hermes DB
Password rotation on the upstream	N/A	Web sign-in immediately picks up the new password; existing app passwords keep working until explicitly revoked
Welcome email	"Click here to set your password"	"Sign in with your organization (AD/LDAP) password"

App passwords are **always Hermes-issued**, regardless of `auth_type`. The upstream directory password is exposed only to the web gate via the LDAP overlay's pass-through bind — never to Dovecot or Nextcloud. See Authentication Settings for the full four-credential architecture and LDAP RemoteAuth for the upstream binding details.

Recipient validation in Postfix

The `recipients` table is queried by Postfix at SMTP time via `mysql:/etc/postfix/mysql-recipients.cf` (mapped to `relay_recipient_maps` in `main.cf`). When a Domain has Recipient Delivery set to `SPECIFIED`, mail arriving for an address **not** in this table is rejected with a `550 User unknown` reply. When Recipient Delivery is `ANY`, the lookup is bypassed for that domain and any recipient is accepted (catch-all).

This is the operational reason to add Relay Recipients **before** flipping a domain to SPECIFIED — flipping first will start rejecting live mail.

Files and containers touched

Path	Owner	Role
`config/hermes/var/www/html/admin/2/view_internal_recipients.cfm`	`hermes_commandbox`	Main page + Edit Options / Edit Encryption / Reset 2FA / Delete modals
`config/hermes/var/www/html/admin/2/add_internal_recipients.cfm`	`hermes_commandbox`	Bulk-add page (local + RemoteAuth + CSV modes)
`config/hermes/var/www/html/admin/2/edit_internal_recipient_backend.cfm`	`hermes_commandbox`	Per-recipient backend override page
`config/hermes/var/www/html/admin/2/inc/get_int_recipient_json.cfm`	`hermes_commandbox`	AJAX hydrator for single-select Edit Options pre-fill

Path	Owner	Role
`config/hermes/var/www/html/admin/2/inc/edit_internal_recipients.cfm`	`hermes_commandbox`	Edit Options handler (+ LDAP cascade on `enforce_mfa` 0→1)
`config/hermes/var/www/html/admin/2/inc/edit_internal_recipients_djigzo.cfm`	`hermes_commandbox`	Edit Encryption handler + cert/keyring queue insertion
`config/hermes/var/www/html/admin/2/inc/delete_internal_recipients.cfm`	`hermes_commandbox`	Per-recipient delete cascade
`config/hermes/var/www/html/admin/2/inc/send_recipient_welcome_email.cfm`	`hermes_commandbox`	Local-auth welcome email (password-reset link)
`config/hermes/var/www/html/admin/2/inc/send_recipient_welcome_email_remoteauth.cfm`	`hermes_commandbox`	RemoteAuth welcome email (org-password sign-in)
`config/hermes/var/www/html/admin/2/inc/ldap_add_user_relay.cfm` / `ldap_add_user_relay_remoteauth.cfm`	`hermes_commandbox`	LDAP stub creation for local / remote auth
`config/hermes/var/www/html/admin/2/inc/ldap_delete_user_relay.cfm`	`hermes_commandbox`	LDAP stub removal on delete
`config/hermes/var/www/html/admin/2/inc/ldap_change_user_access_control.cfm`	`hermes_commandbox`	Group membership swap (one_factor ⇄ two_factor)
`recipients`, `user_settings`, `app_passwords`, `recipient_certificates`, `recipient_keystores`, `cert_generation_queue`, `wblist`, `mailaddr`, `password_reset_requests`, `recipients_temp`	`hermes_db_server`	The recipient-row group + lazy-generation queue
`cn=<user>,ou=users,dc=hermes,dc=local`	`hermes_ldap`	Per-recipient LDAP entry
`cn=relays,ou=groups,dc=hermes,dc=local`	`hermes_ldap`	Relay-recipient group membership
Authelia `totp_configurations` + `webauthn_devices`	`hermes_authelia` storage backend	Cleaned on delete + Reset 2FA Devices
`/etc/postfix/mysql-recipients.cf`	`hermes_postfix_dkim`	Postfix lookup against `recipients` for `relay_recipient_maps`

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

Related

- Domains — relay-domain definitions. Required parent context: a recipient is meaningless without a domain that accepts mail for it. Domain Recipient Delivery `SPECIFIED` is what makes this page's roster authoritative for inbound acceptance.
- Relay Networks — trusted source IPs. The alternative trust path: a source IP listed there can submit outbound without authenticating as a recipient on this page.

- Virtual Recipients — alias-only addresses that forward to a Relay Recipient or external destination. A Virtual Recipient pointing at a deleted Relay Recipient becomes a forwarding hole.
- Relay Host — outbound smarthost. A Relay Recipient that SMTP-AUTHs to send outbound mail still flows through the relay host (if configured) on the way to the Internet.
- LDAP RemoteAuth — required prerequisite for `auth_type='remote'` recipients. Defines the upstream LDAP/AD mappings this page references via `remoteauth_domain`.
- Authentication Settings — full four-credential architecture (web vs IMAP/SMTP vs DAV vs Nextcloud) that recipient app passwords slot into.
- Email Server > Mailboxes — the mail-server-topology equivalent. Don't confuse Relay Recipients (forwarded downstream) with Mailboxes (delivered locally to Dovecot).

Revision #48
Created 2026-05-31 12:52:11 UTC by Dino Edwards
Updated 2026-06-20 13:33:06 UTC by Dino Edwards