

# Aliases

# Aliases

Admin path: **Email Server > Aliases** (`view_mailbox_aliases.cfm`, `inc/add_mailbox_alias_action.cfm`, `inc/edit_mailbox_alias_action.cfm`, `inc/delete_mailbox_alias_action.cfm`, `inc/get_mailbox_alias_json.cfm`).

This page manages **alternate email addresses for local mailboxes** on the Email Server topology. Each row in the `mailbox_aliases` table maps one inbound address (e.g., `sales@company.com`) to either an existing local mailbox or to Postfix's discard transport for silent disposal. The destination must be local — to an existing Dovecot mailbox on this server. For forwarding to external addresses or for relay-topology domains, use [Email Relay > Virtual Recipients](#) instead.

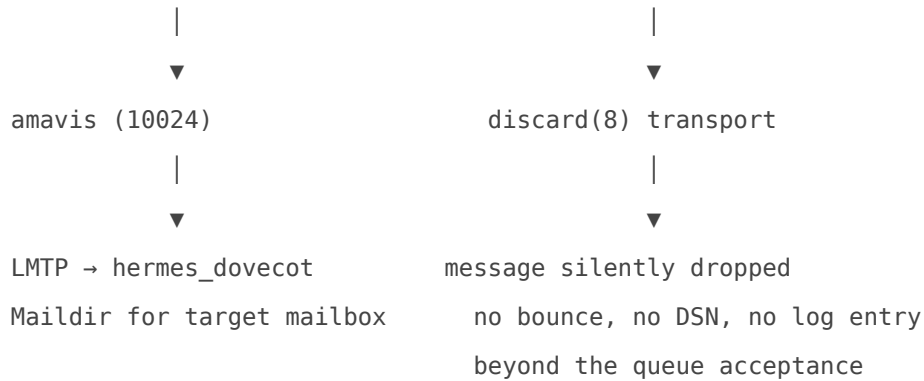
Aliases have **no SMTP authentication, no IMAP/POP3 access, and no password of their own**. They are rewrite rules consumed by Postfix before content filtering. The optional **Send-As** flag adds a row to `sender_login_maps` so the destination mailbox owner can send mail under the alias address from their existing IMAP/Submission session.

## Not the same as Virtual Recipients

Email Server aliases and Email Relay virtual recipients share the same underlying Postfix lookup but enforce different topology rules. See [Virtual Recipients](#) for the full distinction; the short version:

	Mailbox Aliases (this page)	Virtual Recipients
Table	<code>mailbox_aliases</code>	<code>virtual_recipients</code>
Domain type	Mailbox domains ( <code>domains.type = 'mailbox'</code> )	Relay domains ( <code>domains.type = 'relay'</code> or NULL)
Delivery target	A local Dovecot mailbox, or <code>discard:silently</code>	Anywhere — internal or external
UNIQUE on address	Yes (one delivery per alias)	No (fan-out via multiple rows)
Send-As	Optional, surfaced as a toggle	Schema flag, not yet wired through





The MySQL lookup is live — adding a row in this page takes effect on the next inbound message, with no Postfix reload, no `postmap`, and no template regeneration.

## The `mailbox_aliases` table

Column	Type	Role
<code>id</code>	INT PK	Surrogate key
<code>alias_address</code>	VARCHAR(255), <b>UNIQUE</b>	The address being rewritten. Full email only — no catch-all syntax. The UNIQUE constraint enforces one delivery target per alias address.
<code>delivers_to</code>	VARCHAR(255)	Destination. For <code>alias_type = 'forward'</code> this is the local mailbox username; for <code>alias_type = 'discard'</code> this is hardcoded to the literal string <code>discard:silently</code> , which Postfix routes through the <code>discard(8)</code> transport.
<code>alias_type</code>	VARCHAR(20)	<code>forward</code> (default) or <code>discard</code>
<code>send_as</code>	TINYINT(3)	<code>1</code> if the destination mailbox is allowed to send mail as the alias address. Wired into <code>sender_login_maps</code> on insert/update.
<code>domain_id</code>	INT	FK to <code>domains.id</code> ; set on insert from the parsed domain part of <code>alias_address</code> . Used to filter the page by domain and to enforce the mailbox-topology gate.
<code>created_at</code>	DATETIME	Audit timestamp

The UNIQUE key on `alias_address` is the reason fan-out isn't supported here — one inbound address resolves to exactly one destination. To deliver one inbound address to several mailboxes, use a [shared mailbox](#) (which gives multiple users access to a single inbox) or, for true fan-out, use

the relay topology with virtual recipients.

# The two alias types

## Forward

Delivers mail to an existing local mailbox. The mailbox must exist in the `mailboxes` table — the add handler verifies this with `error 16` on failure. The `Delivers To` dropdown is sourced from the live mailbox list (`mailbox_type = 'user'`), so you can only pick a real target.

```
sales@company.com    →  alice@company.com
support@company.com  →  helpdesk@company.com
```

Both addresses must be on a mailbox domain that this server hosts. Cross-domain forwards are allowed as long as both sides are local mailbox domains.

## Discard

Silently drops all mail with no bounce, no DSN, and no error returned to the sender. The handler hardcodes `delivers_to = 'discard:silently'`, which Postfix interprets as the `discard(8)` transport with the literal next hop `silently`. Useful for addresses like `noreply@` or `donotreply@` where bounces would invite spam-mining attempts.

```
noreply@company.com    →  discarded
donotreply@company.com →  discarded
unsubscribe@company.com →  discarded
```

“ **Operational consequence.** Discard is irrecoverable — there is no queue entry, no quarantine, no recovery. The message is accepted by Postfix and immediately dropped. Use discard for addresses that should never receive replies; do not use it as a quiet alternative to bouncing mail you actually want to reject (use Postfix recipient restrictions for that).

## Fields on the page

# Add Alias modal

Field	Notes
<b>Alias Address</b>	Full email. Must validate as an email, must be on a mailbox domain ( <code>domains.type = 'mailbox'</code> ), and must not already exist as a mailbox, an alias, or a virtual recipient. Conflicts produce errors 12 / 13 / 14 / 17 respectively.
<b>Type</b>	<code>Forward (deliver to mailbox)</code> (default) or <code>Discard (silently drop all mail)</code> . JS toggles the Delivers To and Send-As fields based on selection.
<b>Delivers To</b>	Tom Select typeahead populated from <code>mailboxes WHERE mailbox_type = 'user'</code> . Required for forward type, ignored for discard. The handler verifies the target mailbox exists at submit time.
<b>Allow Send-As</b>	<code>No</code> (default) or <code>Yes</code> . Only applies to forward type. When <code>Yes</code> , an <code>INSERT IGNORE</code> into <code>sender_login_maps</code> allows the destination mailbox owner to send under the alias address from their existing Submission session.

# Aliases table

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

Column	Source
Actions	Edit (opens modal) / Delete (opens confirmation modal)
Alias	<code>mailbox_aliases.alias_address</code>
Domain	<code>domains.domain</code> (joined via <code>domain_id</code> )
Type	Badge — <code>Forward</code> (blue) or <code>Discard</code> (dark)
Delivers To	<code>mailbox_aliases.delivers_to</code> for forwards; <code>Silently dropped</code> for discards
Send-As	Badge — <code>YES</code> / <code>NO</code> for forwards; em-dash for discards

A Domain filter dropdown above the table narrows the visible rows to a single mailbox domain. The dropdown only lists domains that currently have at least one alias.

# Edit modal

Address is read-only after creation — changing the local-part would break any send-as mappings that already reference it. Type, Delivers To, and Send-As are all editable, with the same forward/discard toggle behavior as the Add modal. The handler diffs the old send-as state against

the new one and adds or removes the `sender_login_maps` row accordingly so the change to send-as is reflected without rewriting unrelated maps.

## Delete

Per-row delete with a confirmation modal. The handler removes the alias row and any `sender_login_maps` entries for the alias address. Because aliases don't own a Maildir or any on-disk state, deletion is instant and reversible only by re-creating the alias.

# Send-As — what it actually does

When Send-As is enabled on a forward alias, the handler inserts:

```
INSERT IGNORE INTO sender_login_maps (sender, login_user)
VALUES ('sales@company.com', 'alice@company.com');
```

That row participates in Postfix's `smtpd_sender_login_maps` lookup on the submission port. The effect: when `alice@company.com` authenticates to Submission (587) and tries to send a message with `From: sales@company.com`, Postfix accepts the From: because the `(sender, login_user)` pair exists in the map. Without Send-As, Postfix's `reject_sender_login_mismatch` would reject the submission because `alice@` is not the canonical owner of `sales@`.

This makes Send-As a true alternate-identity grant, not just a "vanity From:". The user typically configures the alias as a secondary identity in their mail client (Outlook → Account Settings → multiple email addresses; Apple Mail → Edit Email Addresses; Thunderbird → Manage Identities) and picks it from the From: dropdown when composing.

The deletion handler removes the matching `sender_login_maps` row when the alias is deleted; the edit handler removes the old row and inserts the new one when Send-As is toggled or Delivers To changes.

## Conflict checks at insert time

The add handler runs four duplicate checks before the INSERT:

Check	Error	What it prevents
<code>mailboxes WHERE username = alias_address</code>	13	Alias collides with an actual mailbox. The mailbox itself would always win the lookup, so the alias would be dead weight.

Check	Error	What it prevents
<code>mailbox_aliases WHERE alias_address = alias_address</code>	14	Duplicate alias row (also enforced by the UNIQUE key, but caught earlier with a friendlier message).
<code>virtual_recipients WHERE virtual_address = alias_address</code>	17	Alias collides with a relay-topology virtual recipient. The UNION lookup would return both rows and the resulting fan-out is almost never the intent — the error tells the admin to remove the relay-side row first.
<code>domains WHERE domain = X AND type = 'mailbox'</code>	12	Alias's domain isn't on the mailbox-topology side. Use Virtual Recipients for relay domains.

All four checks are advisory in the UI sense but enforced server-side so a forged form post can't bypass them.

## Domain-delete dependency

There is no explicit dependency check on mailbox-domain deletion for aliases — but mailbox domains are typically not removed unless every mailbox under them is also being removed, and the alias rows become orphaned (`domain_id` no longer resolves) rather than actively harmful. Stale `mailbox_aliases` rows whose `domain_id` no longer exists are skipped by the page query because of the `INNER JOIN domains ... AND d.type = 'mailbox'`. Operational best practice: delete aliases first, then mailboxes, then the domain.

## Failure semantics

What breaks	What happens
Blank alias address in Add	error 10 banner, no DB write
Invalid email format	error 11
Domain not in <code>domains</code> or not mailbox-type	error 12
Address already exists as a mailbox	error 13
Address already exists as an alias	error 14
Address already exists as a virtual recipient	error 17
Forward type with blank Delivers To	error 15
Delivers To target mailbox doesn't exist	error 16

What breaks	What happens
Edit with missing alias_id	error 20
Edit / delete with stale alias_id	error 21
MySQL <code>hermes_db_server</code> down	Postfix <code>virtual_alias_maps</code> lookups fail. Default behavior is to defer affected mail with a temporary error and retry — legitimate mail is held, not bounced.

## Files and containers touched

Path	Owner	Role
<code>config/hermes/var/www/html/admin/2/view_mailbox_aliases.cfm</code>	<code>hermes_commandbox</code>	Page + table + Add / Edit / Delete modals
<code>config/hermes/var/www/html/admin/2/inc/add_mailbox_alias_action.cfm</code>	<code>hermes_commandbox</code>	Add handler with the four-way conflict check
<code>config/hermes/var/www/html/admin/2/inc/edit_mailbox_alias_action.cfm</code>	<code>hermes_commandbox</code>	Edit handler — toggles <code>sender_login_maps</code> on send-as changes
<code>config/hermes/var/www/html/admin/2/inc/delete_mailbox_alias_action.cfm</code>	<code>hermes_commandbox</code>	Delete handler — removes alias row + any send-as map entry
<code>config/hermes/var/www/html/admin/2/inc/get_mailbox_alias_json.cfm</code>	<code>hermes_commandbox</code>	AJAX endpoint that hydrates the Edit modal
<code>/etc/postfix/mysql-virtual.cf</code>	<code>hermes_postfix_dkim</code> (volume-mounted)	The UNION lookup definition shared with <code>virtual_recipients</code>
<code>mailbox_aliases</code> , <code>sender_login_maps</code> , <code>mailboxes</code> , <code>domains</code> , <code>virtual_recipients</code>	<code>hermes_db_server</code>	Storage and conflict-detection tables

Nothing on this page shells out to Postfix — no `postmap`, no `postfix reload`, no template regeneration. The MySQL lookup picks up new rows on the next inbound message.

## Related

- [Email Relay > Virtual Recipients](#) — the relay-topology equivalent. Use that page when the destination is external (Gmail, partner domain) or when fan-out to multiple destinations from one address is needed.
- [Domains](#) — the mailbox-domain list this page filters against. An alias's domain must exist there with `type = 'mailbox'`.
- [Mailboxes](#) — the destination mailbox list. The Delivers To dropdown is populated from active user mailboxes.

- [Shared Mailboxes](#) — when several users need to read the same incoming mail (rather than one user receiving forwards), use a shared mailbox instead of a forward alias.
  - [Mailbox Rules](#) — Sieve-based filtering that runs on the destination mailbox after alias rewrite. Aliases route mail to a mailbox; Sieve rules then sort it within that mailbox.
  - [Settings](#) — the global Email Server toggles. Aliases work regardless of the Mailbox Sharing master switch — they have no Dovecot-side configuration to be gated on.
  - [Authentication Settings](#) — Submission-port authentication that the Send-As flag piggybacks on. A user must be able to authenticate to Submission as their primary address before Send-As lets them switch identities.
- 

Revision #48

Created 2026-05-31 12:52:13 UTC by Dino Edwards

Updated 2026-06-20 13:33:07 UTC by Dino Edwards