Virtual Recipients

Virtual Recipients

Admin path: **Email Relay > Virtual Recipients** (`view_virtual_recipients.cfm`, `inc/addvirtualrecipients.cfm`, `inc/editvirtualrecipient.cfm`, `inc/delete_virtual_recipients.cfm`).

This page manages **forward-only address aliases** on the relay-topology domains configured under Domains. Each row in the `virtual_recipients` table maps one inbound address (or a domain-wide catch-all) to exactly one delivery address. The delivery target can be internal to Hermes, on another relay domain, on a mailbox domain, or anywhere on the public Internet — the row is consumed by Postfix's `virtual_alias_maps` and rewritten at SMTP time, so the forward is transparent to the original sender.

Virtual recipients have **no SMTP authentication, no IMAP/POP3 access, and no password**. They are not user accounts. They are rewrite rules.

Not the same as Mailbox Aliases

The Email Server topology has its own alias page — Email Server > Aliases, backed by the `mailbox_aliases` table — and it serves a different need. The add handler enforces the separation explicitly: trying to add a virtual recipient for a domain flagged as `mailbox` is rejected with the "use Email Server > Aliases" hint.

	Virtual Recipients	Mailbox Aliases
Table	`virtual_recipients`	`mailbox_aliases`
Domain type	Relay domains (`domains.type = 'relay'` or NULL)	Mailbox domains (`mailbox_domains.*`)
Delivery target	Anywhere — internal or external	A local Dovecot mailbox
Resolved by	Postfix `virtual_alias_maps` (MySQL lookup)	Postfix `virtual_alias_maps` (same query, different table)
Auth, IMAP, password	No	No (the resolved mailbox owns those)

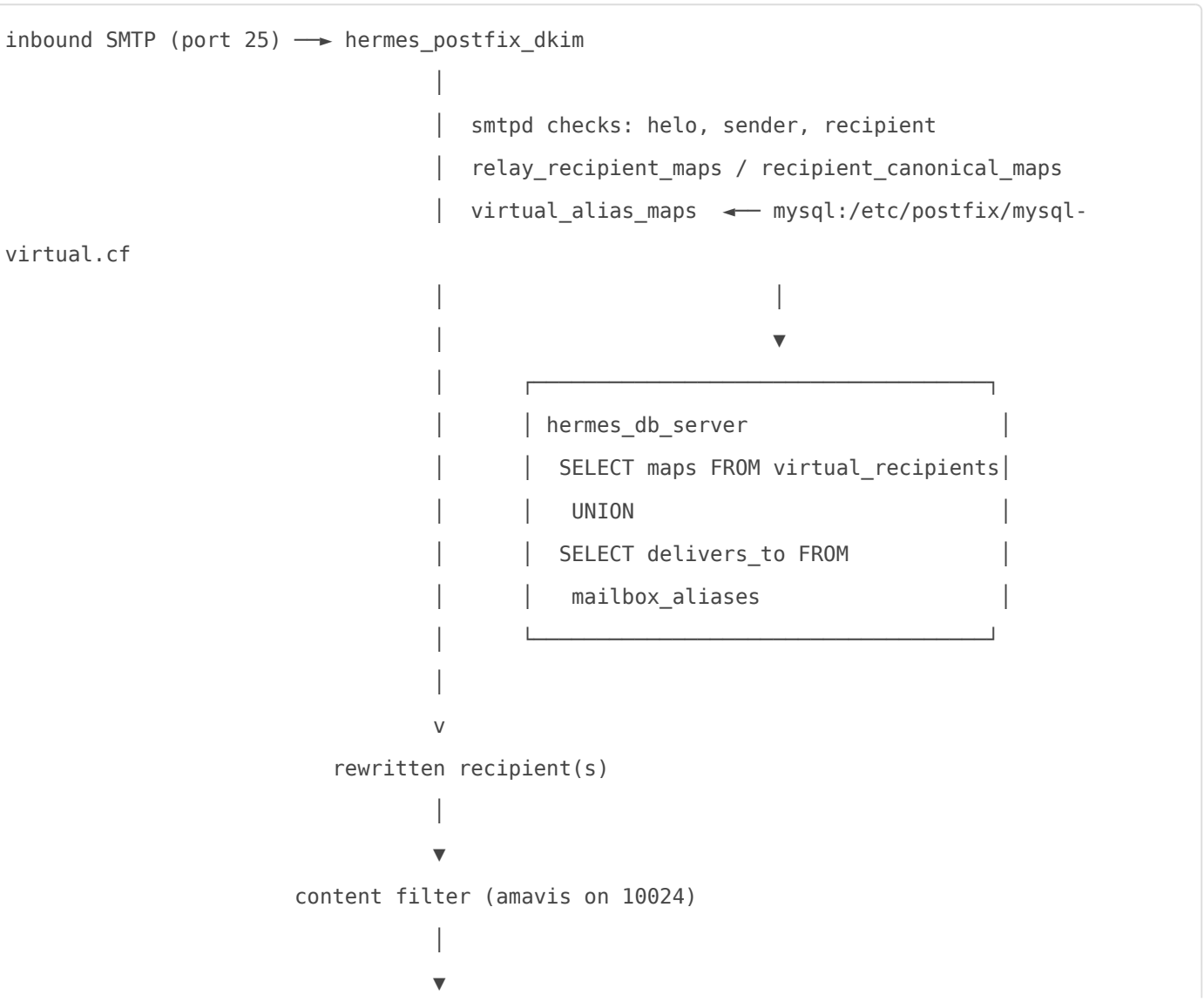
	Virtual Recipients	Mailbox Aliases
Typical use	`info@company.com → admin@company.com, info@externalpartner.example`	`support@company.com → user1@company.com` (where `user1@` is a local mailbox)

The shared `mysql-virtual.cf` lookup is a `UNION` across both tables:

```
SELECT maps          FROM virtual_recipients WHERE virtual_address = '%s'
UNION
SELECT delivers_to   FROM mailbox_aliases    WHERE alias_address   = '%s'
```

Postfix doesn't care which table the answer comes from — but the admin UI separates them so the rule for each topology stays focused.

Storage and lookup path



```
                    outbound or local delivery
```

No file regeneration is required when virtual recipients change. The MySQL lookup is live — adding a row in the admin UI takes effect on the next inbound message, with zero Postfix restart or postmap step. This is the operational reason virtual aliases are stored in MySQL rather than a hash file.

The `virtual_recipients` table

Column	Type	Role
`id`	INT PK	Surrogate key for the row
`virtual_address`	VARCHAR(255)	The address being rewritten. Full email (`info@example.com`) **or** a catch-all token (`@example.com`).
`maps`	VARCHAR(255)	Destination address. Single recipient per row in the current schema.
`alias_type`	VARCHAR(20)	Defaults to `forward`. Reserved for future per-alias behavior flags; not surfaced in the UI today.
`send_as`	TINYINT(3)	Reserved for outbound "send-as" support (allow the destination to send mail as the virtual address). Not wired through Postfix yet.
`policy_id`	INT	Reserved for per-alias Amavis policy attachment. Not surfaced today.
`system`	INT	Provenance marker — `1` = seeded by the install/system-addresses flow (postmaster/abuse/root), `2` = admin-created via this page. The system rows are managed by `update_system_email_addresses.cfm` and recreated when the admin email or postmaster changes.

There is no UNIQUE constraint on `virtual_address` because a single inbound address can fan out to multiple destinations — each destination gets its own row. The add handler dedupes on the `(virtual_address, maps)` pair so the same forward isn't inserted twice.

Two address shapes — specific and catch-all

Specific aliases

A regular forward of one address to one destination:

```
info@company.com       →   owner@company.com
sales@company.com      →   sales-team@externalcrm.example
legal@company.com      →   external-counsel@lawfirm.example
```

The local-part is rewritten by Postfix before content filtering. The recipient never sees the original `info@` / `sales@` / `legal@` address unless the destination mail system surfaces the original envelope.

Catch-alls

A single row starting with `@` matches every local-part on the domain that is **not** already a more specific virtual recipient or a mailbox:

```
@company.com           →   admin@company.com
```

With the catch-all row above, mail to `jdoe@company.com`, `random-string@company.com`, and `does-not-exist@company.com` all forward to `admin@company.com`. Specific aliases on the same domain (`info@company.com → owner@company.com`) win over the catch-all because they match the more specific lookup key first.

Catch-alls are useful for sunset domains, migration phases, or small domains where one mailbox owner is willing to receive everything. They are not appropriate for high-volume domains: every spam attempt against a random local-part lands in the catch-all destination.

Catch-all visibility in the user portal

A user whose mailbox is the **destination** of a catch-all (e.g., `admin@company.com` above) has a special branch in the user portal's Quarantined Messages, Total Messages, and Message History queries. `config/hermes/var/www/html/users/2/index.cfm`, `view_message.cfm`, and `view_message_history.cfm` all consult `virtual_recipients` for catch-all entries that explicitly map TO the logged-in user, then widen the query with a `LIKE '%@domain.tld'` clause so the user sees the

messages that were swept up by the catch-all. Specific aliases do **not** get this treatment yet — a known parity gap for the rare case where one user owns many specific aliases and wants the same widened visibility.

Fields on the page

Add Virtual Recipients card

Field	Notes
Virtual Address(es)	Newline-delimited textarea. Each line is one full email address or a `@domain.com` catch-all. Lowercased, trimmed, deduped against `virtual_recipients` AND `mailbox_aliases` before insert.
Delivers To	Single destination address for the whole batch. Validated as an email. Autocomplete sourced from `inc/getintrecipients.cfm` (existing relay recipients and mailbox addresses) so you can typeahead-pick a known recipient.

The handler iterates the textarea line-by-line and accumulates per-line results. The success banner reports the count and addresses that landed, and separate error banners surface invalid-format lines, lines whose domain isn't configured as a relay domain, lines whose domain is a mailbox domain (with the "use Email Server > Aliases" pointer), and duplicate lines. **No transaction wraps the batch** — partial success is the expected behavior.

Virtual Recipients table

Standard DataTables surface — searchable, sortable, exportable (copy / CSV / Excel / PDF / print), `stateSave: true` so column order and page size persist across reloads. Columns:

Column	Source
Checkbox	Bulk-select for delete
Recipient	`virtual_recipients.virtual_address`
Delivers To	`virtual_recipients.maps`
Actions	Edit (opens modal)

Edit modal

Inline edit of `virtual_address` and `maps`. Re-runs the same domain validation, catch-all detection, and dedupe check as Add — including the rejection of mailbox-domain rows.

Delete

Checkbox-driven bulk delete from the table card. The handler (`delete_virtual_recipients.cfm`) just runs `DELETE FROM virtual_recipients WHERE id = ?` per selected row — there is no dependency check, because nothing else in the schema points back at a virtual recipient row.

Content filter bypass — by design, loud

The yellow callout on the page exists for a reason. Postfix rewrites the recipient **before** the message reaches Amavis content filtering, but Amavis policy lookups key on the **post-rewrite** recipient. If the destination address is an external Internet address (Gmail, Outlook.com, a personal mailbox, etc.), Amavis applies the default outbound policy to it — which typically means lighter spam/banned-files enforcement than a domain-scoped inbound policy would.

The net effect: mail aliased through a virtual recipient to an external address is generally **less aggressively filtered** than the same mail delivered to a local mailbox or relayed to a known partner domain. This is fine for legitimate forwards, but admins who use virtual recipients to bridge a sunset domain to a personal Gmail should expect Amavis to be permissive about it. Tighten the policy by editing the destination recipient's `recipients` row directly under Relay Recipients if the destination is itself a known Hermes recipient.

Domain-delete dependency

Deleting a relay domain via Domains is blocked when virtual recipients reference it. `deletedomain.cfm` runs:

```
SELECT * FROM virtual_recipients WHERE virtual_address LIKE '%<domain>%'
```

Any match aborts the domain delete with error code 2 and the admin must clear the matching rows from this page before the domain can be removed. The same back-pressure protects against silently stranding a forward when its destination domain disappears.

System-managed rows

A few rows in `virtual_recipients` are created and managed by the **System > Server Setup** flow, not by this page directly:

Pattern	Created by
`postmaster@<every-domain>` → admin email	`inc/update_system_email_addresses.cfm` on every Server Setup save
`root@<every-domain>` → admin email	Same
`abuse@<every-domain>` → admin email	Same

These rows are marked `system = '1'` (the install/system flow) versus admin-created rows which are marked `system = '2'`. Editing or deleting a system-managed row from this page works mechanically, but the row will be recreated on the next Server Setup save. Edit the admin email there if you want a different destination for these reserved local-parts; do not maintain them by hand here.

Failure semantics

What breaks	What happens
Virtual address blank in Add	error 1 banner, no DB write
Delivers To blank or invalid email in Add	error 2/3 banner, no DB write
Edit virtual address fails email or catch-all format	`session.m = 10`, redirect, no DB write
Edit Delivers To blank or invalid	`session.m = 11` / `12`, redirect, no DB write
Domain not in `domains` table	`session.m = 13` on edit; per-line invalid-domain banner on add — line skipped, others continue
Domain is a mailbox domain	Per-line invalid-domain banner with the "use Email Server > Aliases" hint; line skipped
Duplicate `(virtual_address, maps)` pair in `virtual_recipients` or `mailbox_aliases`	Per-line duplicate banner on add; `session.m = 14` on edit
Delete with no rows selected	`session.m = 1` banner, no DB write
MySQL `hermes_db_server` down	Postfix `virtual_alias_maps` lookups fail. By default Postfix defers mail to the affected recipients with a temporary error and retries on the next queue run; legitimate mail is held, not bounced.

Bulk import

The current page supports newline-delimited paste into the Add textarea, which is the practical bulk path: paste hundreds of `alias@domain.com` lines (all forwarding to one destination) at once, click Add, get a per-line outcome report. A separate CSV import is not provided because the table is intentionally one-destination-per-row — fan-out is expressed by adding the same `virtual_address` multiple times with different `maps`, which is easier to do in the textarea than in a CSV.

Files and containers touched

Path	Owner	Role
`config/hermes/var/www/html/admin/2/view_virtual_recipients.cfm`	`hermes_commandbox`	Page + Add card + table + modals
`config/hermes/var/www/html/admin/2/inc/addvirtualrecipients.cfm`	`hermes_commandbox`	Add handler with per-line validation
`config/hermes/var/www/html/admin/2/inc/editvirtualrecipient.cfm`	`hermes_commandbox`	Edit handler
`config/hermes/var/www/html/admin/2/inc/delete_virtual_recipients.cfm`	`hermes_commandbox`	Delete handler (per selected id)
`config/hermes/var/www/html/admin/2/inc/getintrecipients.cfm`	`hermes_commandbox`	Autocomplete source for the Delivers To field
`config/hermes/var/www/html/admin/2/inc/update_system_email_addresses.cfm`	`hermes_commandbox`	Manages the `system = '1'` rows (postmaster/root/abuse)
`/etc/postfix/mysql-virtual.cf`	`hermes_postfix_dkim` (volume-mounted)	Postfix MySQL lookup definition for `virtual_alias_maps`
`virtual_recipients`, `mailbox_aliases`, `domains`	`hermes_db_server`	The lookup tables and the domain-type gate

Nothing on this page shells out to Postfix — there is no postmap, no `postfix reload`, no template regeneration. The MySQL lookup is the only integration surface.

Related

- Domains — the relay-topology domain list these aliases attach to. Domain deletes are blocked when virtual recipients still reference the domain.
- Relay Recipients — recipient validation for domains with Recipient Delivery = SPECIFIED. A specific relay recipient and a virtual recipient can coexist for the same address; the relay recipient wins for recipient-list validation, the virtual recipient still rewrites at delivery.

- Email Server > Aliases — the mailbox- topology equivalent. Aliases for domains where Hermes is the destination MTA live there.
- Email Server > Shared Mailboxes — when several users need to read the same incoming mail (not just one user receiving forwards), use a shared mailbox instead of a fan-out virtual recipient.
- Server Setup — manages the `system = '1'` postmaster/root/abuse forwards. Change the admin email there to retarget those reserved local-parts.

Revision #8
Created 2026-05-31 12:52:12 UTC by Dino Edwards
Updated 2026-05-31 14:01:13 UTC by Dino Edwards