Scheduled Tasks

Scheduled Tasks

Admin path: **System > Scheduled Tasks** (`view_scheduled_tasks.cfm`, `inc/ofelia_generate_config.cfm`, `inc/run_scheduled_task_action.cfm`, `inc/toggle_ofelia_job_action.cfm`, `inc/restart_ofelia.cfm`).

This page is the admin surface over **Ofelia**, Hermes's cron runner. Ofelia (`mcuadros/ofelia:latest`) sits next to the application containers, mounts the Docker socket, and on a schedule does `docker exec <container> <command>` for each configured job. The page lists every job in the `ofelia_jobs` table, displays its humanized schedule and last manual-run timestamp, and exposes per-row **Enable/Disable** and **Run Now** controls.

Hermes does not use the host's crond. Every recurring task — certificate renewal, the daily update check, quarantine notifications, mail-queue health checks, DMARC report processing, malware-feed refresh, log rotation — runs through this single Ofelia container and is manageable from this page.

Why Ofelia and not host cron

A traditional host crontab does not fit Hermes's deployment model:

Requirement	Host cron problem	Ofelia behavior
Run a command inside `hermes_commandbox` or `hermes_dmarc` on a schedule	Host cron has to `docker exec` from outside; failure modes (missing container, wrong user) surface in syslog, not in the admin UI	Ofelia speaks Docker natively; jobs are `job-exec` blocks against a named container
Notify the admin when a job fails	Cron emails the local UNIX user; meaningless inside a container deployment	Ofelia has a built-in SMTP notifier that emails `admin_email` via `hermes_postfix_dkim:10026` (the auto-DKIM-signing re-injection port) when `mail-only-on-error = true`
Survive a host reboot the same way every other Hermes service does	Cron units have to be packaged separately	`hermes_ofelia` is just another container in `docker-compose.yml`; `restart: unless-stopped` covers it

Requirement	Host cron problem	Ofelia behavior
Be inspectable and runnable on demand from the web UI	Out-of-band; admin would need shell access	This page reads the same table Ofelia reads and can re-fire any job synchronously

The trade-off is that `config.ini` is regenerated from the database — so direct hand-edits to `/etc/ofelia/config.ini` are **overwritten on every save**. The DB is the source of truth.

How a scheduled job flows through the stack

```
+-----------------------+
| ofelia_jobs (MariaDB) |    <-- canonical source of truth
+-----------+-----------+
            |
            | Save / Toggle / install --apply-schema
            v
+-------------------------------------------------------+
| inc/ofelia_generate_config.cfm                        |
|   1. SELECT * FROM ofelia_jobs WHERE active = '1'     |
|   2. Render /opt/hermes/tmp/<tok>_ofelia_jobs         |
|   3. dos2unix (CRLF safety)                           |
|   4. Read /opt/hermes/conf_files/ofelia_config.ini    |
|        (template with POSTMASTER_EMAIL,               |
|         ADMIN_EMAIL, OFELIA_JOBS_GO_HERE markers)     |
|   5. REReplace each marker with live values           |
|   6. Move final file to /etc/ofelia/config.ini        |
|   7. cfinclude restart_ofelia.cfm                     |
+----------------------------+--------------------------+
                             |
                             v
+-------------------------------------------------------+
| hermes_ofelia container                               |
|   reads /etc/ofelia/config.ini on start               |
|   fires `docker exec <container> <command>` on        |
|   each job's schedule, capturing stdout/stderr        |
|   on failure: emails admin_email via 10026            |
```

```
+-----------------------------------------------------+
```

Configuration storage

Table	Role
`ofelia_jobs`	One row per scheduled job
`scheduled_job_runs`	Append-only history of **manual** Run Now invocations from this page; Ofelia's own scheduled executions are not recorded here

`ofelia_jobs` schema (relevant columns):

Column	Type	Notes
`job_name`	`varchar(255)`	The **full bracketed header** as Ofelia consumes it, e.g. `[job-exec "hermes-quarantine-notify"]`. The display-friendly name shown in the table is the text between the quotes (the page extracts it with a regex).
`schedule`	`varchar(255)`	Ofelia format — either 6-field cron (`sec min hr dom mon dow`), 5-field cron, or `@every <duration>` (e.g. `@every 60s` , `@every 10m`, `@every 1h`)
`command`	`varchar(255)`	The shell command Ofelia runs inside the container
`container`	`varchar(255)`	Target container — `hermes_commandbox` for most jobs, `hermes_dmarc` for DMARC report processing, `hermes_mail_filter` for fangfrisch
`active`	`int(11)`	`1` **= enabled,** `2` **= disabled.** Disabled jobs stay in the DB but are filtered out of the generated `config.ini`.
`no_overlap`	`tinyint(3)`	When `1`, Ofelia emits `no-overlap = true` so a still-running invocation prevents the next tick from firing. Used for short-interval jobs (`@every 60s` cert-queue, quarantine-notify).
`type`	`varchar(255)`	Category tag for grouping (`certbot`, `hermes`, `dmarc`, `pushover`, `malware_feeds`, `system`)

The seeded job set

A fresh install (`hermes_install.sql`) seeds these jobs. All start enabled.

Job	Schedule	Container	What it does
`renew-acme-certificate`	Daily 12:05	`hermes_commandbox`	Runs certbot renew across all ACME-issued certs; reloads dependent services on success
`hermes-message-cleanup`	Daily 01:30	`hermes_commandbox`	Enforces `msgs` retention policy (Pro: per-policy; Community: global)
`hermes-update-check`	Daily 04:30	`hermes_commandbox`	Polls GitHub Releases; writes the cache file the dashboard reads. See System Update § Daily update check.
`acme-validate-ip`	Every 30 min	`hermes_commandbox`	Refreshes mailbox-domain SAN cert state when the gateway's public IP changes
`hermes-health-check-mailqueue`	Every 15 min	`hermes_commandbox`	Pushover alert when `mailq` count exceeds the threshold
`hermes-dmarc-report`	Daily 02:30	`hermes_dmarc`	Fetches DMARC RUA reports, parses them into the `opendmarc` DB
`hermes-authelia-log-rotate`	Daily 02:00	`hermes_commandbox`	Rotates Authelia's access logs
`hermes-quarantine-notify`	Every 60s, `no-overlap`	`hermes_commandbox`	Issues quarantine-release emails to recipients with pending messages
`hermes-process-cert-queue`	Every 60s, `no-overlap`	`hermes_commandbox`	Drains the encryption cert lookup queue for outbound S/MIME / PGP recipients
`hermes-fangfrisch-refresh`	Every 10 min	`hermes_mail_filter`	Refreshes third-party ClamAV signature feeds (SecuriteInfo, Sanesecurity, etc.)

New jobs added by later features (signature-map regen for the body milter, the post-upgrade hook caller, etc.) appear here automatically as they are seeded into `ofelia_jobs`. The page renders whatever is in the table — there is no hardcoded job list in the CFML.

The page columns

The DataTable renders one row per `ofelia_jobs` row.

Column	What it shows
Name	The display-friendly name (text between the quotes in `job_name`)
Type	The `type` category tag
Schedule	Humanized form — `@every 60s` becomes "Every 60 seconds", `0 30 04 * * *` becomes "Daily at 04:30", `0 0 02 * * *` becomes "Daily at 02:00", and so on. Hover for the raw cron expression (commit `8e954d1d`). Anything the humanizer can't cleanly parse falls through to the raw string.
Container	Target container (`hermes_commandbox`, `hermes_dmarc`, `hermes_mail_filter`, ...)
Command	The literal command Ofelia runs
Status	Bootstrap-switch toggle (Enabled / Disabled), AJAX-driven
Last Run (manual)	Most recent **Run Now** click from this page; Ofelia's own scheduled fires do not write here
Actions	The **Run Now** button

Enable / Disable toggle

The switch posts to `inc/toggle_ofelia_job_action.cfm` with the `job_name` and `new_state` (`1` or `2`). The handler:

1. Looks up the row; rejects if not found.
2. `UPDATE ofelia_jobs SET active = ?`.
3. Re-runs `ofelia_generate_config.cfm`, which writes a fresh `config.ini` containing only the enabled rows.
4. Restarts `hermes_ofelia` via `restart_ofelia.cfm`.
5. On any failure during step 3 or 4, **rolls the `active` flag back** and returns the error in JSON. The UI reverts the switch and surfaces the error.

The transactional behavior matters — a half-applied state where the DB says "disabled" but Ofelia is still running the job is exactly the confusing situation an admin would not be able to diagnose from this page.

The JS layer surfaces a confirm prompt before disabling jobs on a **critical list** (`renew-acme-certificate`, `hermes-update-check`, `hermes-process-cert-queue`, `hermes-quarantine-notify`). The backend trusts the request — admins with web access already have the means to disable everything via direct SQL if they want to. The prompt is a guard against an accidental click, not an authorization gate.

Run Now

The button posts to `inc/run_scheduled_task_action.cfm`, which executes the job's `command` synchronously and returns JSON with status, duration, exit code, and output (capped at 2048 bytes for the DB history, full body in the response). The result is displayed in a modal with a spinner-then-summary view.

Three execution strategies, picked from the command shape:

Command shape	Strategy
`/usr/bin/curl --silent http://localhost:8888/schedule/<name>.cfm`	Routed via `cfhttp` for clean body capture. This is the majority of Hermes jobs — the actual work is implemented as a CFML schedule script and Ofelia is just a trigger.
`container != hermes_commandbox`	Proxied via `cfexecute docker exec <container> <command>`. Used for `hermes-dmarc-report` (targets `hermes_dmarc`) and `hermes-fangfrisch-refresh` (targets `hermes_mail_filter`).
Anything else inside `hermes_commandbox`	`cfexecute` directly — the page itself runs inside `hermes_commandbox`, so this is equivalent to what Ofelia would do.

Hard cap on the manual-trigger path is **300 seconds**. Ofelia's own scheduled runs have no such cap; if a job legitimately needs to run longer, scheduled execution is fine but Run Now will time out.

Every Run Now invocation appends a row to `scheduled_job_runs` — including failures, including runs of disabled jobs (the page allows firing a disabled job on demand without re-enabling it). The Last Run column reads from this table.

> **By design.** Run Now and the schedule run independently. Firing a job manually does **not** reset Ofelia's next-scheduled-fire clock. If you Run Now a job that is also scheduled to fire in 30 seconds, it will fire again 30 seconds later — for the `no-overlap` jobs, Ofelia will skip the scheduled fire if the manual run is still in progress; for the others, both runs will happen.

The config.ini template

`config/hermes/opt/hermes/conf_files/ofelia_config.ini` is a small placeholder file:

```
[global]
smtp-host = hermes_postfix_dkim
smtp-port = 10026
email-to = ADMIN_EMAIL
email-from = POSTMASTER_EMAIL
mail-only-on-error = true

OFELIA_JOBS_GO_HERE
```

`ofelia_generate_config.cfm` does three `REReplace` passes against this template — `ADMIN_EMAIL` and `POSTMASTER_EMAIL` from `system_settings`, `OFELIA_JOBS_GO_HERE` from the rendered `[job-exec ...]` blocks — and writes the result to `/etc/ofelia/config.ini`. The intermediate work happens under `/opt/hermes/tmp/<customtrans3>_*` with a final atomic `move` into place, which is also why a partial regen does not leave the live file half-written.

When direct edits to config.ini are appropriate

There are exactly two situations where editing `/etc/ofelia/config.ini` directly makes sense:

1. **Debugging Ofelia itself** — flipping `mail-only-on-error` to `false` so every successful run notifies, or adding `verbose = true` to the global block to flood `docker logs hermes_ofelia` with detail.
2. **Adding a one-shot job that you don't want in the DB** — e.g., a migration script that should run once at the next scheduled time.

In both cases, the change survives until the next save on this page or the next install-script run. If you need a persistent custom job, add it to `ofelia_jobs` directly via SQL and the regen will pick it up.

Failure semantics

What breaks	What happens
`ofelia_jobs` is empty	The page shows a warning callout; Ofelia generates no jobs and idles. Re-run `install_hermes_docker.sh --apply-schema` to re-seed.
Toggle handler fails mid-regen	`active` flag rolled back to its previous value; switch reverts in the UI; error surfaced. Live `config.ini` is unchanged.
`restart_ofelia.cfm` fails (container missing, Docker socket gone)	Toggle response carries the error message; live `config.ini` is the new one but Ofelia hasn't reread it yet. Manual `docker compose restart hermes_ofelia` recovers.
Run Now times out (>300s)	`cfexecute` raises; the JSON response is `success: false` with an exception; `scheduled_job_runs` still gets the failure row.
Run Now command exits non-zero	Modal shows the stderr in the output pane; the row still inserts into `scheduled_job_runs` with `exit_code` set to whatever the process returned.
Ofelia's own scheduled run fails	Ofelia emails `admin_email` via 10026 (auto-DKIM-signed). **Not** reflected in the Last Run column on this page — that column is manual-only.
`dos2unix` not installed inside `hermes_commandbox`	Regen aborts with `error.cfm` traceback. The shipped image has it; only relevant for custom builds.

Files and containers touched

Path	Owner	Role
`config/hermes/var/www/html/admin/2/view_scheduled_tasks.cfm`	`hermes_commandbox`	The page (renders the table, hosts the toggle + Run Now JS)
`config/hermes/var/www/html/admin/2/inc/run_scheduled_task_action.cfm`	`hermes_commandbox`	Run Now AJAX endpoint
`config/hermes/var/www/html/admin/2/inc/toggle_ofelia_job_action.cfm`	`hermes_commandbox`	Enable/Disable AJAX endpoint
`config/hermes/var/www/html/admin/2/inc/ofelia_generate_config.cfm`	`hermes_commandbox`	Config regenerator — reads `ofelia_jobs`, writes `config.ini`
`config/hermes/var/www/html/admin/2/inc/restart_ofelia.cfm`	`hermes_commandbox`	`docker container restart hermes_ofelia` wrapper
`config/hermes/opt/hermes/conf_files/ofelia_config.ini`	`hermes_commandbox`	Template with `ADMIN_EMAIL` / `POSTMASTER_EMAIL` / `OFELIA_JOBS_GO_HERE` markers
`config/ofelia/config.ini`	`hermes_ofelia` (live)	Regen target
`ofelia_jobs` table	`hermes_db_server` (`hermes` DB)	Canonical job list

Path	Owner	Role
`scheduled_job_runs` table	`hermes_db_server` (`hermes` DB)	Manual-run history
`/var/run/docker.sock` (host mount → `hermes_ofelia`)	host filesystem	How Ofelia issues `docker exec` against other containers

Future work

- **Inline schedule editing** — today, schedule + command edits happen on feature-specific pages (e.g., the Malware Feeds settings page edits `hermes-fangfrisch-refresh`'s schedule). A "create new job" and inline edit on this page is planned for a later release.
- **External job triggers via API** — issues #222 (Hermes Internal API) and #223 (API tokens) will eventually let external systems POST to `/api/scheduled-tasks/<name>/run` with a token, replacing the web-UI-only Run Now flow. Not yet built.
- **Surface Ofelia's scheduled-run history** — `scheduled_job_runs` records manual runs only because that is what the page writes. Ofelia's own per-run history sits in `docker logs hermes_ofelia` and is not currently tabled. A future enhancement could parse Ofelia's stdout into a similar history table.

Related

- System Update — the `hermes-update-check` job is the daily GitHub Releases poll that drives the dashboard's update-available cell
- DNS Resolver — most scheduled jobs depend on outbound DNS resolution flowing through `hermes_unbound`
- System Certificates — the `renew-acme-certificate` job is what actually keeps Let's Encrypt certs current; the page only registers and binds them
- System Settings — `admin_email` (Ofelia failure notification target) and `postmaster` (sender) are both read from here at config regen
- System Status — dashboard cells reflect outputs that several of these jobs produce (mail queue, update status)
- Storage Topology — `hermes_ofelia` is stateless; its config lives in the Config tier (`config/ofelia/`)

Revision #48
Created 2026-05-31 12:52:01 UTC by Dino Edwards
Updated 2026-06-20 13:33:00 UTC by Dino Edwards