Console Settings

Console Settings

Admin path: **System > Console Settings** (`view_console_settings.cfm`, `inc/get_console_settings.cfm`, `inc/edit_console_settings.cfm`, `inc/generate_auth_nginx_configuration.cfm`, `inc/generate_nginx_configuration.cfm`, `inc/generate_authelia_configuration.cfm`, `inc/generate_nextcloud_configuration.cfm`, `inc/edit_ciphermail_settings.cfm`, `preload_restart_nginx.cfm`).

This page configures **how the outside world reaches the Hermes web console** — the FQDN or IP that nginx terminates TLS on, the certificate it presents, and three HTTPS hardening toggles (HSTS, OCSP stapling, OCSP stapling verify). It is the single source of truth for the console hostname; every other component that needs to know "where do I live" (Authelia session cookie, Nextcloud trusted domain and theming URL, the User Console link in Nextcloud's top bar, Ciphermail portal redirect URL, OIDC discovery URI) is regenerated from this page when the Console Address changes.

Pairs with Server Setup, which configures the gateway's mail-side identity (Postfix `myorigin` / `myhostname` and the host IP). The two pages together define every name Hermes presents to the world.

Where the console host fits

```
Browser  ──►  hermes_nginx (443)
                  │ server_name = <Console Address>
                  │ ssl_certificate = <Console Certificate>
                  ▼
              auth_request /authelia
                  │
                  ▼
              hermes_authelia
                  │ session.cookies[].domain = <Console Address>
                  │ authelia_url             = https://<Console Address>/authelia
```

```
                ▼
hermes_commandbox (admin + user portal)
hermes_nextcloud  (NC trusted_domain + theming URL +
                   user_oidc discovery URI +
                   External Sites "User Console" link)
hermes_ciphermail (portal URL = Console Address)
```

Every one of those downstream consumers is rewritten from the value saved on this page. Direct edits to `auth.conf`, `hermes-ssl.conf`, `configuration.yml`, Nextcloud's `config.php`, the Ciphermail portal URL, or OIDC discovery are **overwritten on the next save**.

Configuration storage

Both the Console Address and the four hardening / cert settings live in the `parameters2` table with `module = 'console'`. The page is wired strictly against that table — there are no file-backed secrets here, only DB values.

Setting	`parameters2.parameter`	Default
Console Address (IP or FQDN)	`console.host`	`smtp.domain.tld` (seed)
Console Certificate (FK into `system_certificates.id`)	`console.certificate`	`29` (seed snakeoil)
DH parameters	`console.dhparam`	`enable`
HSTS	`console.hsts`	`enable`
OCSP Stapling	`console.ssl_stapling`	`enable`
OCSP Stapling Verify	`console.ssl_stapling_verify`	`enable`

> **DH parameters note.** The `console.dhparam` row is still in the schema and still set by the form handler when a DH file exists, but commit `2dbc2bd3` ("ECDHE-only ciphers, remove DH parameters feature") moved the active TLS cipher suite to ECDHE-only — DH is no longer offered. The setting is therefore inert; leave it at the default.

Fields on the page

Console Address (IP or FQDN)

The hostname or IP nginx terminates TLS on for `/admin`, `/users`, `/nc`, `/portal` (Ciphermail), and every other console-served path. Accepts:

- **IPv4** — validated against the standard dotted-quad regex
- **IPv6** — validated against the bracketed/colon form
- **FQDN** — validated by the email-trick (`IsValid("email", "bob@<host>")`)

`edit_console_settings.cfm` trims whitespace and strips any trailing zone-file dots (`mail.example.com.` becomes `mail.example.com`) before saving. That stripping happens at the input boundary so every downstream consumer — `autoconfig.cfm`, `autodiscover.cfm`, nginx vhost generation, the NC theming URL, the OIDC discovery URI — sees an identical canonical string. Outlook for Mac is one of several MUAs that breaks on the trailing dot, hence the strip.

> **If you set Console Address to an IP** and then the server's IP changes, you must update **both** Console Address (this page) **and** Host IP Address (Server Setup) — they are stored in separate parameters and neither cascades to the other. The page surfaces this in a warning callout.

Console Certificate

Free-text autocomplete that searches `system_certificates` via `getcertificates.cfm` (an ajax endpoint). Selecting a row populates a hidden `certificateno_1` field with the certificate's row ID, plus five read-only display fields (subject, issuer, serial, type, friendly name). The handler validates the ID exists in `system_certificates` before saving — an empty or unknown ID falls through to the next step with `step = 3`, which means the existing `console.certificate` value is preserved.

The selected cert becomes `nginx`'s `ssl_certificate` / `ssl_certificate_key` for every console-facing vhost. Certificate upload, renewal, and Let's Encrypt are managed on System Certificates; this page is the **binding** of one of those certificates to the console hostname.

HSTS, OCSP Stapling, OCSP Stapling Verify

Three boolean (`enable` / `disable`) selects. Each is substituted into `/opt/hermes/templates/hermes-ssl.conf` at regen time:

Toggle	Effect on the generated `hermes-ssl.conf`

HSTS	`add_header Strict-Transport-Security "max-age=31536000; preload"` (enabled) vs. the same line commented out (disabled)
OCSP Stapling	`ssl_stapling on;` (enabled) vs. `#ssl_stapling on;` (disabled)
OCSP Stapling Verify	`ssl_stapling_verify on;` (enabled) vs. `#ssl_stapling_verify on;` (disabled)

Defaults are all `enable` and should stay that way for any publicly-reachable console. Disable only if you have a specific reason (e.g., HSTS preload conflict during a hostname migration window).

Save flow — the cascade

Clicking **Save & Apply Settings** posts `action=edit`, which runs `edit_console_settings.cfm` as a strict 7-step sequence. Each step gates on the previous step's success (`<cfif step is "N">`) — any validation failure short-circuits with `cflocation url="#cgi.http_referer#"` and `session.m` set to the matching alert code; no partial state lands.

```
step 1  Validate + write console.host
step 2  Validate + write console.certificate
step 3  (DH param — inert)              ──► step 4
step 4  Write console.hsts
step 5  Write console.ssl_stapling
step 6  Write console.ssl_stapling_verify
step 7  Regen + restart cascade  ──┐
                                   │
        generate_auth_nginx_configuration.cfm   (rewrites snippets/auth.conf)
        generate_nginx_configuration.cfm        (rewrites snippets/hermes-ssl.conf)
        generate_authelia_configuration.cfm     (rewrites authelia/configuration.yml)
        generate_nextcloud_configuration.cfm    (rewrites nc/config.php trusted_domains)
        occ user_oidc:provider Hermes_SEG       (discovery URI + end-session URI)
        occ config:app:set external sites       (NC top-bar "User Console" link JSON)
        occ theming:config url                  (NC theming URL)
        edit_ciphermail_settings.cfm            (Ciphermail portal URL)
        restart_authelia.cfm                    (preload-style restart)
        restart_ciphermail.cfm                  (preload-style restart)
        preload_restart_nginx.cfm               (last — see below)
```

`preload_restart_nginx.cfm` is the canonical Hermes pattern for restarting the proxy from inside a request that is **served by the proxy**. A plain `docker container restart hermes_nginx` would close the request's own connection and the browser would see `ERR_CONNECTION_REFUSED` on the redirect

back. The preload page returns a full HTML response that includes a `fetch()` to a separate `restart_nginx_post.cfm` endpoint and a poll-loop that waits for nginx to come back before redirecting to `view_console_settings.cfm`. Always use this pattern from any handler that ends in an nginx restart.

The Nextcloud `occ` calls in steps 7d–7f are all wrapped in `<cftry>...<cfcatch type="any"></cfcatch></cftry>` and marked **non-fatal** in the comments. A Nextcloud container that is down or slow at the moment of save will leave the NC-side values stale; on the next save (or a manual `occ` invocation) they will catch up.

> **By design.** The cascade is destructive — there is no dry-run, no diff preview, no "stage changes." Saving rewrites all four config files and restarts three containers. Plan saves outside business hours if the deployment is busy.

Operational consequence — changing the Console Address mid-flight

A live Console Address change is the single most disruptive operation on this page. While the cascade runs (typically 30–60 seconds end to end including container restarts):

Surface	Behavior during the change
The admin page that initiated the change	Held by `preload_restart_nginx.cfm` until nginx returns 200 on `/index.cfm`, then redirected back
Other open admin sessions	Will see `502 Bad Gateway` for the nginx restart window; their session cookie is also now scoped to the **old** hostname and they will be re-prompted to log in after they reload at the new address
User portal / Nextcloud / Webmail sessions	Same — all session cookies are domain-scoped; users at the old hostname must navigate to the new one and re-authenticate
Mail flow (SMTP/IMAP/Submission)	**Unaffected.** Postfix and Dovecot do not depend on the console nginx vhost.
Outbound DKIM signing	Unaffected.

Surface	Behavior during the change
Webmail OIDC	Discovery URI is rewritten at step 7d but the change only takes effect after Nextcloud picks up the new `occ user_oidc` settings — in practice this is instant because `occ` writes synchronously

If the new Console Address has no DNS record yet, the change still saves (Hermes does not DNS-resolve the value) but every external client request will fail until DNS catches up.

Bypassing this page — risks

There are three other paths that can change the console hostname or hostname-derived values **without** going through this cascade. Each one leaves Hermes in an inconsistent state. Do not use them unless you are recovering from a broken cascade and you know what you are doing.

1. **Direct edit of `parameters2`** — sets `console.host` but does not regenerate `auth.conf`, `hermes-ssl.conf`, `configuration.yml`, `config.php`, theming, External Sites, OIDC, or Ciphermail.
2. **Direct edit of `config/nginx/.../snippets/*.conf` or `config/authelia/configuration.yml`** — the next save on this page overwrites your hand-edits.
3. **A future Hermes CLI Management Console** (`scripts/hermes-cli.sh`) is planned but not yet built. It will expose Change Console Host as a menu option so admins have a recovery path when a bad Console Address change has locked them out of the web UI. Until it ships, the only recovery is direct SQL + manual regen-script invocations against the `hermes_commandbox` container.

Per-domain nginx vhosts are NOT regenerated by this page

This page rewrites `snippets/auth.conf` and `snippets/hermes-ssl.conf` — the global console snippets. **Per-domain vhosts** generated for mailbox domains, autodiscover, autoconfig, and any other domain-scoped surface live in separate templates and are rendered on their own pages (Mailboxes > Domains, mostly).

If you edit one of those per-domain templates by hand and expect already-generated vhosts to pick it up, they will not. Either re-render each affected domain from its own UI, or run the appropriate domain-regen include directly. The same rule applies in reverse — a console hostname change does **not** rewrite per-domain server blocks that were generated before the change. Most installs do not need to, because per-domain vhosts use the domain hostname, not the console hostname. If a

per-domain vhost was unusually wired to the console hostname (manual customisation), re-render it.

Failure semantics

What breaks	What happens
Console Address validation fails (invalid IPv4/IPv6/FQDN)	`session.m = 3`, redirect, no DB write
Console Certificate ID not found in `system_certificates`	`session.m = 2`, redirect, no DB write
Nginx config syntax error after template substitution	`nginx -t` fails inside `restart_nginx_post.cfm`; the previous live config stays loaded (nginx never gets the `reload`), but the on-disk file is the broken one. Recovery: fix the template, re-save.
Authelia container fails to start after `configuration.yml` regen	See Authentication Settings § Failure semantics. The `restart_authelia.cfm` output is logged but not surfaced in the success banner.
Nextcloud `occ` calls error out	Logged silently (cftry wrapping); next save retries.
Ciphermail not running	The portal URL stays out of sync; next save catches up after the container is back.

Files and containers touched

Path	Owner	Role
`config/hermes/var/www/html/admin/2/view_console_settings.cfm`	`hermes_commandbox`	Page
`config/hermes/var/www/html/admin/2/inc/edit_console_settings.cfm`	`hermes_commandbox`	Save handler (7-step cascade)
`config/hermes/var/www/html/admin/2/inc/get_console_settings.cfm`	`hermes_commandbox`	Load handler
`config/hermes/var/www/html/admin/2/preload_restart_nginx.cfm`	`hermes_commandbox`	Restart-and-redirect overlay
`config/hermes/opt/hermes/templates/hermes-ssl.conf`	`hermes_commandbox`	Console nginx server-block template
`config/hermes/opt/hermes/templates/auth.conf`	`hermes_commandbox`	Console auth_request snippet template
`config/hermes/opt/hermes/templates/configuration.yml`	`hermes_commandbox`	Authelia config template
`/etc/nginx/snippets/hermes-ssl.conf`	`hermes_nginx`	Live console TLS / hardening snippet (regen target)

Path	Owner	Role
`/etc/nginx/snippets/auth.conf`	`hermes_nginx`	Live console auth_request snippet (regen target)
`/config/configuration.yml`	`hermes_authelia`	Live Authelia config (regen target)
`/var/www/html/config/config.php`	`hermes_nextcloud`	Live NC config — `trusted_domains` updated (regen target)
`oc_appconfig` (appid `external`, configkey `sites`)	`hermes_nextcloud` MariaDB	Top-bar User Console link JSON blob
`oc_appconfig` (appid `theming`, configkey `url`)	`hermes_nextcloud` MariaDB	NC theming URL
`user_oidc` provider `Hermes_SEG`	`hermes_nextcloud`	OIDC discovery + end-session URIs

Every cross-container call uses `docker exec` per the standard Hermes pattern. The temp-shell-script convention (`/opt/hermes/tmp/<token>_*.sh`) is used for the External Sites `occ` call because the JSON value has quoting/escaping that `cfexecute`'s `arguments` parsing handles unreliably; writing a small shell script and executing it instead of passing the JSON inline avoids that whole class of bug.

Related

- Server Setup — the mail-side server identity (Postfix `myorigin` / `myhostname`, Host IP). Companion to this page; the two together define every name Hermes presents.
- System Certificates — uploading, renewing, and managing the certificates this page selects from
- Authentication Settings — Authelia configuration; this page rewrites its config file as part of every save
- SMTP TLS Settings — the mail-side TLS certificate binding, the analogue of "Console Certificate" for SMTP banners
- DNS Resolver — if the Console Address is an internal-only FQDN, this page's resolver settings determine whether other Hermes containers can reach it

---

Revision #14
Created 2026-05-31 12:51:56 UTC by Dino Edwards
Updated 2026-06-13 12:30:00 UTC by Dino Edwards