File Extensions

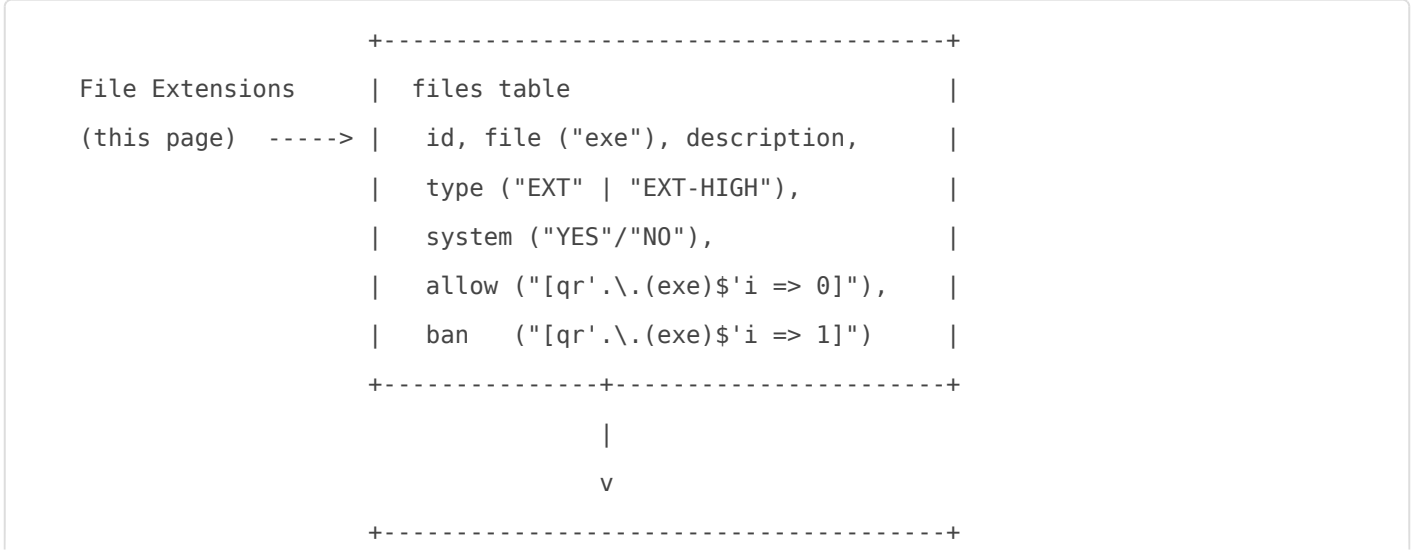
File Extensions

Admin path: **Content Checks > File Extensions** (`view_file_extensions.cfm`, `inc/get_file_extensions.cfm`, `inc/update_amavis_config_files.cfm`).

This page maintains the catalogue of **attachment file extensions** that Amavis can match on. Each entry is a single extension such as `.exe`, `.docm`, or `.iso` paired with a description and a sensitivity flag (Standard vs. High Risk). The page itself does not block anything — it only registers extension candidates. The block / allow decision is taken by a File Rule that bundles extensions into a named ruleset, which is then applied to recipients via an SVF policy on Anti-Spam Settings. File Extensions is the building-block page; File Rules and SVF Policies are where the ruleset is composed and bound to traffic.

The extension catalogue ships with a system-managed list of common high-risk types (`.exe`, `.scr`, `.pif`, `.com`, `.bat`, `.vbs`, `.js`, `.jar`, `.ps1`, and dozens more) that cannot be deleted from the UI. Operators add custom extensions on top — typically Office macro-enabled types in environments that don't allow macros, archive formats they want to surface separately, or new attack-surface file types as they appear in the wild.

Where File Extensions sits



```
|  File Rules                          |
|   bundle extensions into named       |
|   rulesets with per-extension        |
|   allow / ban / priority             |
+--------------+-----------------------+
               |
               v
+--------------------------------------+
|  Anti-Spam Settings (SVF Policies)   |
|   bind a File Rule to recipient(s)   |
+--------------+-----------------------+
               |
               v
+--------------------------------------+
|  Amavis 50-user.HERMES               |
|   @banned_filename_re emitted per    |
|   rule on every save chain           |
+--------------------------------------+
```

Amavis enforces the resulting `@banned_filename_re` regex sets at content-filter time inside `hermes_mail_filter`. A matched extension triggers Amavis's `final_banned_destiny` action (`D_BOUNCE`, `D_DISCARD`, or `D_PASS` — set globally on Anti-Spam Settings).

What "matched" means in Amavis

The stored allow / ban snippets are case-insensitive regexes anchored to the end of the filename:

```
[qr'.\.(exe)$'i => 1]      (ban; case-insensitive)
[qr'.\.(exe)$'x  => 0]     (allow; case-sensitive)
```

This means:

- `invoice.exe` matches `.exe`
- `Invoice.EXE` matches `.exe` (because the `i` modifier is set by default on Add)
- `invoice.pdf.exe` matches `.exe` (the *trailing* extension is the one Amavis tests)
- `invoice.exe.pdf` does **not** match `.exe` — it matches `.pdf`, and the trailing-extension rule is the only one that fires

The double-extension confusion case (`invoice.pdf.exe`) is the historic reason this list exists. Amavis sees the real trailing extension; the user sees only the displayed-name prefix and a familiar icon.

The page

A page guide callout, an Add Extensions card with a bulk textarea, a Custom File Extensions DataTable (editable / deletable), and a Read-Only System File Extensions DataTable (the shipped list).

Add File Extensions card

Field	Stored as	Notes
File Extensions	`files.file` + `files.description`	One per line; format `.ext description`. The leading dot is stripped on save (so the row stores `exe`, not `.exe`); the description is auto-prefixed with `(.ext)` so the DataTable shows `(.docm) Microsoft Word Macro-Enabled Document` regardless of how the operator typed it
Extension Type	`files.type`	`EXT` (Standard) or `EXT-HIGH` (High Risk). Purely a classification tag for the UI badges — Amavis treats both the same
Case Sensitivity	drives which template is rendered into `files.allow` / `files.ban`	Insensitive (default, recommended) uses `_insense` templates with the `i` regex modifier; sensitive uses `_sense` templates with `x` only — for environments where you want `.EXE` to differ from `.exe`

The handler line-splits the textarea on either LF or CRLF, strips whitespace, validates each entry, and inserts the valid ones. Per entry it checks:

- The extension starts with `.`
- The extension matches `^[.][a-zA-Z0-9\-\.\_]+$` (alphanumeric, dash, period, underscore — nothing else)
- The description is non-blank (required)
- No row with the same `file` already exists in the `EXT` / `EXT-HIGH` type space (a `.docm` cannot exist as both Standard and High Risk)

Each rejected line is collected into a per-row error list that surfaces in the partial-success alert; the valid entries still insert. The `(.ext)` prefix on the description is auto-prepended so the catalogue stays self-describing regardless of how the operator typed the row.

Custom File Extensions DataTable

Column	Source
(checkbox)	Selection for bulk Delete Selected
Extension	`.<files.file>` (the leading dot is displayed in the UI even though it isn't stored)
Description	`files.description`
Actions	Per-row Delete button (single-row confirm)

The DataTable shows only rows with `system = 'NO'` and excludes `type = 'CUSTOM-EXPRESSION'` rows (those belong to File Expressions, which uses the same `files` table with a different `type` discriminator).

System File Extensions DataTable (read-only)

The shipped catalogue — every row from `files` where `system = 'YES'` and `type IN ('EXT', 'EXT-HIGH')`. These rows are filtered out of every DELETE path on this page (`AND system = 'NO'` is part of every DELETE query). The UI gives them no checkbox and no Delete button; attempting a forged POST that targets a system row surfaces alert `m = 11` and is rejected.

Standard rows get an "Info" badge, High Risk rows get a "Danger" badge. The badge is cosmetic — Amavis treats both the same as banned-extension candidates once they're wired into a File Rule.

Foreign-key guard on delete

A custom extension cannot be deleted while it is referenced by any File Rule. The single-row Delete handler runs:

```
SELECT COUNT(*) AS cnt FROM file_rule_components
WHERE file_id = :id
```

If `cnt > 0`, the delete is refused with alert `m = 10` and the DataTable shows the offending rule name(s) ("This file extension is used in the following File Rule(s): **HighRisk-block**"). The operator's path is to open File Rules, remove the extension from the rule, then come back here and delete it.

Bulk Delete applies the same guard per-id and accumulates partial results — the success alert reports "N deleted, M skipped" with the skipped rows' rule names attached so the operator knows exactly what to unwire first.

Save and apply flow

```
1. View page submits action="add_entries" | "delete" | "bulk_delete"
2. For each valid entry:
     a. Read the case-sensitive/insensitive allow + ban templates
        from /opt/hermes/scripts/file_allow_{sense|insense} and
        file_deny_{sense|insense}
     b. Substitute THE-EXTENSION placeholder with the (dot-stripped)
        extension name
     c. INSERT INTO files (file, description, type, system, allow, ban)
3. If at least one row was added or deleted:
     a. update_amavis_config_files.cfm:
          - Read /opt/hermes/conf_files/50-user.HERMES (template)
          - Substitute SERVER-NAME, SERVER-DOMAIN, sa-spam-subject-tag,
            final-virus-destiny, final-banned-destiny, final-spam-destiny,
            final-bad-header-destiny, enable-dkim-verification,
            enable-dkim-signing placeholders from spam_settings
          - Render every File Rule's components into an
            @banned_filename_re block (per-rule, in priority order,
            using the allow/ban regex stored on each files row)
          - Substitute HERMES-USERNAME / HERMES-PASSWORD from
            /opt/hermes/creds/ for the Amavis MySQL lookup
          - Back up /etc/amavis/conf.d/50-user -> 50-user.HERMES,
            move rendered file into place
     b. docker exec hermes_mail_filter /etc/init.d/amavis force-reload
        (30-second timeout)
4. session.m = 1 (add) | 2 (single delete) | 12 (bulk delete)
```

Amavis is reloaded with `force-reload` rather than restarted — the daemon re-reads `50-user` without dropping connections, and mail in flight is not interrupted. The full container restart that Anti-Spam Settings and Score Overrides trigger is not needed here because no SpamAssassin state is being touched.

The reload step is wrapped in `cftry`/`cfcatch` with **comment "Log but don't block — extensions were added"** — if the reload itself fails, the DB rows are already in place and the next save (or manual `force-reload`) will re-render. The page does not roll back on reload failure.

Failure semantics

Alert	Trigger
`m = 1`	Add Extensions completed (with `entries_added` / `entries_skipped` / `entry_errors` set on session for the per-row breakdown alert)
`m = 2`	Single Delete succeeded; Amavis reloaded
`m = 10`	Single Delete refused — the extension is wired into at least one File Rule (rule names surfaced in the alert)
`m = 11`	Attempt to delete a system row (`system = 'YES'`) — refused at the DB query
`m = 12`	Bulk Delete completed (with `bulk_deleted` / `bulk_skipped` / `bulk_errors` set on session)
`m = 30`	Add submitted with an empty textarea

The per-row error list is HTML-rendered into the alert body so the operator sees every rejection at once ("Must start with dot: foo", "Invalid characters: .x@y", "Description required: .docm", "Duplicate: .exe"). No row is silently dropped without an explanation in the alert.

Files and containers touched

Path	Owner	Role
`config/hermes/var/www/html/admin/2/view_file_extensions.cfm`	`hermes_commandbox`	The page (validation + bulk add + DataTables + Amavis reload)
`config/hermes/var/www/html/admin/2/inc/get_file_extensions.cfm`	`hermes_commandbox`	Loads custom + system rows for the two DataTables
`config/hermes/var/www/html/admin/2/inc/update_amavis_config_files.cfm`	`hermes_commandbox`	Renders `50-user` from template + File Rules (called on every change)
`config/hermes/opt/hermes/scripts/file_allow_insense` / `file_allow_sense`	`hermes_commandbox`	Allow-regex templates with `THE-EXTENSION` placeholder
`config/hermes/opt/hermes/scripts/file_deny_insense` / `file_deny_sense`	`hermes_commandbox`	Ban-regex templates with `THE-EXTENSION` placeholder
`config/hermes/opt/hermes/conf_files/50-user.HERMES`	`hermes_commandbox` (read) -> `hermes_mail_filter` (live `/etc/amavis/conf.d/50-user`)	Canonical Amavis template; receives the rendered `@banned_filename_re` blocks
`/etc/amavis/conf.d/50-user`	`hermes_mail_filter`	Live Amavis config; reloaded with `force-reload` on every save
`files` table, `type IN ('EXT','EXT-HIGH')`	`hermes_db_server` (`hermes` DB)	Source of truth for the catalogue (system + custom)
`file_rule_components` table	`hermes_db_server` (`hermes` DB)	Cross-reference checked by the delete guard

Path	Owner	Role
`hermes_mail_filter` container	—	Hosts Amavis; receives `force-reload` (not restart) on every change

Related

- File Expressions — sibling page for full regex patterns against any filename (not just extension); rows live in the same `files` table under `type = 'CUSTOM-EXPRESSION'`
- File Rules — bundles extensions and expressions into named, prioritised rulesets; the consumer of every row this page creates
- Message Rules — content-level SpamAssassin rules (header / body / regex) — the body / header equivalent of what File Extensions does for attachment names
- Anti-Spam Settings — defines `final_banned_destiny` (what Amavis does with a banned-extension match) and binds File Rules to recipients via SVF Policies
- Antivirus Settings — ClamAV runs in the same Amavis pass; a virus verdict on the same attachment overrides the banned-extension result
- Score Overrides — sibling Amavis tuning page; both write into Amavis configuration but extension blocks are categorical (matched -> banned) where SA rules are weighted
- Perimeter Checks — none of this matters for connections that never make it past the SMTP-time perimeter
- Message History — a banned-extension rejection appears with Type `Banned` and the matched extension surfaced in the detail view
- System Logs — Amavis logs the matched regex as `Blocked BANNED (.exe,.bat,...)` on the `amavis[...]:` line

Revision #48
Created 2026-05-31 12:52:24 UTC by Dino Edwards
Updated 2026-06-20 13:33:13 UTC by Dino Edwards