

The files that Drupal does not **track**

Those files carry a real cost in storage, governance, and trust. File Inspector helps you handle them. →

Storage and the database drift apart in every system

Two stores must stay in agreement: the bytes in storage, and the database that should track them.

The moment both exist, they drift. Records get deleted while the files stay. Files get uploaded but never registered. Jobs fail halfway through.

These unreferenced files are **untracked files**, a form of **dark data**: kept by the organization, but no longer used. Every content system meets this sooner or later.

Untracked files cost more than disk space

The organization stays responsible for files it cannot even see.

STORAGE AND GOVERNANCE

Files accumulate with no record and no owner. Disk grows, but the real issue is that **nothing accounts for them.**

EXPOSURE AND SECURITY

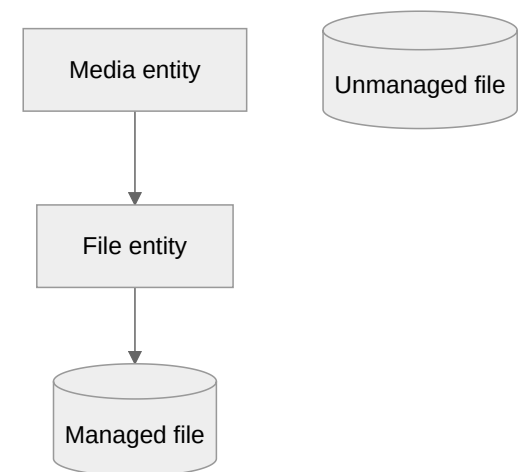
An untracked file stays in storage, downloadable by anyone with the URL and shown on no admin screen. The risk is minor for an ordinary asset, but **serious for sensitive data** like legal documents or internal reports.

Drupal sees a file only through its records

A **file entity** is Drupal's record of a file. A **media entity** wraps that file with a type, metadata, and a reusable identity, which is Drupal's recommended way to handle files. Usage tracking records where each file is used.

A file exists for Drupal only through these records.

With no record, the bytes are something **no part of the application owns.**



Two kinds of untracked file

Both are files nothing references, whether or not Drupal still has a record.

UNMANAGED FILES

Bytes on disk with **no file entity at all**. Drupal is blind to them. They arrive by direct upload, FTP, migration, or backup.

ORPHANED MANAGED FILES

The file entity exists, but usage dropped to zero after the node or media was deleted. Drupal knows the file, it just **never cleans it up**.

Existing tools, and where they stop

At best, they make Drupal aware of the file without making it useful.

FANCY FILE DELETE

It removes unmanaged files and detached entities.
Developer-oriented and deletion-only.

AUDIT FILES

The closest relative. It cross-checks storage, entities, and usage, and can register a file, but only as a **bare file entity**.

The solution

File Inspector inventories every file, then lets you **delete or reuse** each one.

File Inspector is built to close that gap

The principles on which the File Inspector module rests

- **Visibility first:** start with a readable inventory of files and their status, because **nothing unseen can be governed**.
- **Reconcile, not just delete:** pull useful files into management as reusable **Media**, not a bare entity an editor cannot place.
- **Built for content owners:** clear reports and actions in the UI, with filters, bulk operations, and proper permissions.

Configure the module

A settings form keeps the inventory focused, so the report shows **signal, not noise**. Two switches decide whether delete and import to Media are offered.

- **Allowed MIME types:** inspect only the types worth reviewing, images, PDFs, Office documents. Wildcards like `image/*` work, everything else is skipped.
- **Excluded folders:** leave out asset and system folders such as `css`, `js`, and `styles`. They hold generated files, not content.
- **Embedded web folders:** identify the folders that hold a flipbook or microsite to avoid reporting every supporting asset.

Generate the report

File Inspector walks the entire file system with **generators**, keeping one file in memory at a time, and processes them in **batches** so a large run never times out. Results land in a dedicated table.

File System Tracker

Total files tracked:2890
Unprocessed: 0
Managed:2356
Unmanaged:529
Imported: 0
Deleted: 0
Web embedded:5

This operation will scan the file system and add all files to the tracker table.

Scan the file system and add files to the tracker.

Process Tracked Files

All tracked files have been processed.

This operation will check managed status for all tracked files and update their records.

Check managed status for tracked files.

Track files

Process tracked files

Read the report: overview and file list

One summary table counts every file by status, **managed, unmanaged**, or imported, with a grand total. The statuses that need attention stand out, and each row links straight into that filtered list.

File Inspector ☆			
Overview		View Files	
Status	Count	Description	Actions
Unprocessed	0	Files discovered but not yet analysed.	View files
Managed	2356	Files referenced by a Drupal entity.	View files
Unmanaged	529	Orphaned files not referenced by any entity.	View files
Imported	0	Files imported into the media library.	View files
Deleted	0	Files removed from the file system.	View files
Web embedded	5	Files belonging to embedded web content.	View files
Total	2890		

Filter and act

The inventory is built on **Views**, so it filters by status, MIME type, and date, runs bulk operations, and extends without code.

File name

Status

Unprocessed

Managed

Unmanaged

Imported

Deleted

Web embedded

Hold Ctrl (Cmd on Mac) and click to select more than one status.

MIME Type

- Any -

Sort by

Track created

Order

Desc

Filter

Action: - Select -

Apply to selected items

[Show all columns](#)

☐	ID	File Name	MIME Type	Status	Media	Actions
☐	256	demo-image-16_9_12.jpg	image/jpeg	Unmanaged	—	Import ▼
☐	1	RM1.png	image/png	Unmanaged	—	Import ▼
☐	257	demo_translation_13.pdf	application/pdf	Unmanaged	—	Import ▼

Reconcile each file

You decide whether to delete a file or import it into a media item.

Many unmanaged files are live assets that were never registered. **One action** wraps the file in a Media item, file entity included, with the media type chosen at import. It lands in the **Media library** ready to use.

One action, both entities.

File Information

File Name	Path	MIME Type	Size
demo-icon_3.png	public://default-content/demo-icon_3.png	image/png	11.8 KB

Media Bundle *

Image

Select which media bundle to use for importing this file.

This operation will create a managed File entity and wrap it in a Media entity, making it available in the media library

This action cannot be undone.

Import File

Cancel

Storage drift is universal, and now manageable

Files routinely end up in storage with **no entity managing them**, a blind spot with real cost in storage, governance, and trust. Existing tools proved the need but stop at deletion or a bare file entity.

File Inspector inventories first, reconciles all the way to a usable **Media item**, is built for content managers, and scales to real sites. **A user-friendly way to manage untracked files in Drupal.**



Is your Drupal storage a blind spot?

At **Metadrop** we build and maintain Drupal modules like File Inspector, so untracked files become a UI task.

Follow us for more on Drupal, content, and file management.



Save



Share



Comment

