AssetConnector

File: `config/core/asset-connector.php`
Version: 10.01
Type: Singleton PHP class
Purpose: One unified class to upload ANY file to `crm_assets` from anywhere in the system.

Why It Exists

Before AssetConnector, every module that needed to upload files (Assets Manager, AirBlog, AirMail, FotoTrafiX API, Company Management logo upload, AI image generation) had its own copy of the upload logic — 300+ lines of metadata extraction, S3 upload, DB insert, and collection linking, each with slightly different path construction, error handling, and validation. This caused recurring bugs (wrong paths, missing thumbnails, broken S3 keys) and made maintenance painful.

AssetConnector centralizes all of this into a single class with one main method: `store()`.

Quick Start

```
// Get the singleton
$connector = AssetConnector::getInstance($mysqli);

// Upload from browser form ($_FILES)
$result = $connector->storeUpload('file', $companyId, $siteId, $userId);

// Upload from a local file path
$result = $connector->storeFile('/tmp/photo.jpg', $companyId, $siteId,
$userId);

// Upload from binary content (API response, AI-generated, etc.)
$result = $connector->storeContent($binaryData, 'image.png', $companyId,
$siteId, $userId);
```

All three return the same format:

```
// Success
[
    'success'  => true,
    'asset_id' => 447,
```

```
    'asset'    => [
        'id'              => 447,
        'company_id'      => 10,
        'site_id'         => 21,
        'filename'        => 'my_photo_6613a4b2e1f3c.jpg',
        'original_filename' => 'My Photo.jpg',
        'file_path'       =>
'https://s3.us-west-1.wasabisys.com/pilot-air4.media/companies/copa/protected
/sites/21/assets/2026/03/my_photo_6613a4b2e1f3c.jpg',
        'thumbnail_path'  =>
'https://s3.us-west-1.wasabisys.com/pilot-air4.media/companies/copa/public/si
tes/21/assets/2026/03/thumbs/thumb_6613a4b2e1f3c.jpg',
        's3_key'          =>
'companies/copa/protected/sites/21/assets/2026/03/my_photo_6613a4b2e1f3c.jpg'
,
        's3_key_thumb'    =>
'companies/copa/public/sites/21/assets/2026/03/thumbs/thumb_6613a4b2e1f3c.jpg
',
        'file_size'       => 2456789,
        'mime_type'       => 'image/jpeg',
        'asset_type'      => 'image',
        'width'           => 4032,
        'height'          => 3024,
        'title'           => 'Sunset at Venice Beach',
        'collection_id'   => 12,
    ],
]

// Failure
[
    'success' => false,
    'error'   => 'File too large. Max: 50 MB, got: 72.3 MB',
]
```

The `store()` Method — Full Options

```
$result = $connector->store([
    // ── SOURCE (one required) ──
    'file_key'     => 'logo',           // $_FILES key
    'local_path'   => '/tmp/file.jpg',  // absolute path on server
    'content'      => $binaryData,      // raw file content

    // ── REQUIRED ──
```

```
    'company_id'   => 10,                // company that owns this asset
    'site_id'      => 21,                // site association
    'user_id'      => 7,                 // who uploaded it

    // ── FILE INFO (optional) ──
    'filename'     => 'custom.png',      // required with 'content', optional
otherwise
    'original_name'=> 'My File.png',     // display name (auto-detected if
omitted)

    // ── METADATA OVERRIDES (optional) ──
    'title'        => 'Company Logo',    // overrides EXIF title
    'description'  => 'Our brand logo', // overrides EXIF description
    'tags'         => 'logo,branding',   // MERGED with EXIF tags (not
replaced)
    'category'     => 'branding',        // asset category
    'copyright'    => '© 2026 Copa',     // overrides EXIF copyright
    'creator'      => 'John Smith',      // overrides EXIF creator
    'alt_text'     => 'Copa logo',       // accessibility text

    // ── S3 STORAGE (optional) ──
    'module'       => 'assets',          // S3 path segment: assets, airblog,
airmail, etc.
    'visibility'   => 'protected',       // 'public' or 'protected' (default:
protected)
    'subfolder'    => 'logos',            // extra path segment in S3 key

    // ── COLLECTION (optional) ──
    'collection'   => 'Company',         // name (auto-creates album) or int
(collection_id)
    'collection_type' => 'album',        // type for auto-created collections

    // ── PORTAL (optional) ──
    'portal_visible'   => false,         // show in AirPortal client view
    'download_enabled' => true,          // allow portal download
    'account_id'       => null,          // CRM account FK
    'project_id'       => null,          // CRM project FK

    // ── PROCESSING (optional) ──
    'skip_metadata'      => false,       // skip EXIF/IPTC extraction
    'skip_thumbnail'     => false,       // skip thumbnail generation
    'skip_transcription' => true,        // skip Whisper transcription
(default: true)
```

```
    // ── VALIDATION (optional) ──
    'max_size'       => 10485760,       // override max file size (bytes)
    'allowed_types'  => ['jpg','png'],  // override allowed extensions
]);
```

Supported File Types

Type	Extensions	Max Size
Image	jpg, jpeg, png, gif, webp, svg, bmp, tiff, tif, ico	50 MB
Video	mp4, mov, avi, webm, mkv, m4v, wmv, flv, mpg, mpeg	500 MB
Audio	mp3, wav, aac, ogg, flac, m4a, wma	100 MB
Document	pdf, doc, docx, xls, xlsx, ppt, pptx, txt, csv, rtf, odt, zip, gz, etc	100 MB

What It Does Automatically

1. File Validation

- Checks file exists and uploaded without error
- Validates extension against allowed types
- Enforces size limits (per-type defaults or custom override)

2. Metadata Extraction (via MediaProcessor)

- **Images:** EXIF, IPTC, XMP — title, description, keywords, copyright, creator, camera make/model, capture date, ISO, aperture, focal length, GPS coordinates
- **Videos:** Duration, dimensions, codec info + optional Whisper transcription + AI keyword extraction
- **Documents/Audio:** MIME type detection only
- All metadata stored in `crm_assets.exif_data` as JSON
- Search content auto-generated from title + description + tags + transcript

3. Thumbnail Generation (via MediaProcessor)

- **Images:** JPEG at 400px width, quality 85
- **Videos:** WebP at 400px width, quality 65
- **SVG:** No thumbnail needed (renders at any size)
- **Documents/Audio:** No thumbnail

4. S3 Upload (via StorageService)

- Original file → `protected/` prefix (or `public/` if specified)
- Thumbnail → `public/` prefix (always accessible)
- Path convention: `companies/{slug}/{visibility}/sites/{siteId}/{module}/YYYY/MM/{filename}`
- Storage quota enforced (increments `companies.storage_used_bytes`)
- Thumbnails don't count toward quota

5. Database Insert

- Full record in `crm_assets` with all metadata columns
- Both `file_path` (full URL) and `s3_key` (raw key) stored
- `processing_status` set to `completed`
- `storage_provider` set to `wasabi`

6. Collection Linking (optional)

- Pass `'collection' => 'Company'` to auto-find or create an album
- Pass `'collection' => 42` to link to existing collection by ID
- Auto-sets cover image if collection has none
- Uses `crm_asset_collection_items` junction table with `display_order`

Convenience Methods

`storeUpload($fileKey, $companyId, $siteId, $userId, $extra = [])`

Shorthand for `$_FILES` uploads:

```
$result = $connector->storeUpload('file', 10, 21, 7, [
    'collection' => 'Company',
    'visibility' => 'public',
]);
```

`storeFile($localPath, $companyId, $siteId, $userId, $extra = [])`

Shorthand for server-side files:

```
$result = $connector->storeFile('/var/www/air4.media/tmp/export.pdf', 10, 21,
7, [
```

```
    'category' => 'exports',
    'skip_metadata' => true,
]);
```

`storeContent($content, $filename, $companyId, $siteId, $userId, $extra = [])`

Shorthand for binary data:

```
// AI-generated image
$result = $connector->storeContent($dalleResponse, 'ai-generated.png', 10,
21, 7, [
    'title' => 'AI Generated: sunset beach',
    'tags' => 'ai,generated,dalle',
    'collection' => 'AI Generated',
]);
```

`ensureCollection($name, $companyId, $siteId, $userId, $type = 'album')`

Get or create a collection by name. Returns collection ID:

```
$collectionId = $connector->ensureCollection('Company', 10, 21, 7);
// Returns existing ID if "company" slug exists, creates new album otherwise
```

`delete($assetId)`

Delete an asset and all its S3 files:

```
$result = $connector->delete(447);
// Deletes: original from S3, thumbnail from S3, DB record
// Cascade: removes from crm_asset_collection_items
// Decrements: companies.storage_used_bytes
```

Usage Examples

Company Logo Upload

```
require_once $resolver->resolve('config/core/asset-connector.php');
$connector = AssetConnector::getInstance($mysqli);

$result = $connector->storeUpload('logo', $companyId, $siteId, $userId, [
    'collection'     => 'Company',
```

```
    'visibility'     => 'public',
    'title'          => $companyName . ' Logo',
    'tags'           => 'logo,branding,company',
    'category'       => 'branding',
    'skip_metadata'  => true,
    'skip_thumbnail' => false,
    'allowed_types'  => ['jpg', 'jpeg', 'png', 'gif', 'svg', 'webp'],
    'max_size'       => 5 * 1024 * 1024, // 5MB
]);

if ($result['success']) {
    // Update company record with the S3 URL
    $stmt = $mysqli->prepare("UPDATE companies SET logo_path = ? WHERE id =
?");
    $logoUrl = $result['asset']['file_path'];
    $stmt->bind_param('si', $logoUrl, $companyId);
    $stmt->execute();
}
```

Blog Post Image

```
$result = $connector->storeUpload('featured_image', $companyId, $siteId,
$userId, [
    'module'     => 'airblog',
    'visibility' => 'public',
    'collection' => 'Blog Images',
    'title'      => $postTitle . ' — Featured Image',
]);
```

FotoTrafiX Batch Import

```
foreach ($files as $filePath) {
    $result = $connector->storeFile($filePath, $companyId, $siteId, $userId,
[
        'collection'         => $eventName,
        'skip_transcription' => true,
    ]);
    if (!$result['success']) {
        logError("Import failed for {$filePath}: " . $result['error']);
    }
}
```

Email Attachment

```
$result = $connector->storeContent($attachmentData, $attachmentName,
$companyId, $siteId, $userId, [
    'module'   => 'airmail',
    'category' => 'email-attachment',
    'subfolder' => 'attachments',
]);
```

Architecture

```
AssetConnector (singleton)
├── resolveSource()        → normalize file from $_FILES / path / binary
├── detectAssetType()      → image / video / audio / document / other
├── extractMetadata()      → MediaProcessor (EXIF/IPTC/XMP)
├── generateThumbnail()    → MediaProcessor (ImageMagick/FFmpeg)
├── uploadToS3()           → StorageService → WasabiApi
├── insertRecord()         → crm_assets INSERT
├── resolveCollection()    → find or create crm_asset_collections
└── linkToCollection()     → crm_asset_collection_items INSERT
```

Dependencies

Class	File	Role
`StorageService`	`config/core/storage-service.php`	S3 path construction, upload, quota
`WasabiApi`	`config/core/wasabi-api.php`	Low-level S3 SDK wrapper
`MediaProcessor`	`config/core/media-processor.php`	EXIF extraction, thumbnail generation
`PathResolver`	`config/core/path-resolver.php`	Resolve file paths

Database Tables

Table	Role
`crm_assets`	Main asset records
`crm_asset_collections`	Albums, galleries, catalogs
`crm_asset_collection_items`	Junction table (asset ↔ collection)
`companies`	Storage quota tracking (`storage_used_bytes`)

Table	Role
`company_storage_log`	Storage change audit trail

Error Handling

- All errors are caught and returned as `['success' => false, 'error' => '...']`
- Errors are logged via `logError()`
- Temp files are always cleaned up, even on failure
- Non-fatal errors (metadata extraction, thumbnail generation) are logged as warnings but don't fail the upload
- S3 upload failure IS fatal — no orphaned DB records

S3 Path Convention

```
companies/{company-
slug}/{visibility}/sites/{siteId}/{module}/YYYY/MM/{filename}
                          │                          │
                          │                          └─ assets, airblog,
airmail, etc.
                          └─ public (thumbnails, public assets)
                             protected (originals, private assets)

Thumbnails:
companies/{company-
slug}/public/sites/{siteId}/{module}/YYYY/MM/thumbs/{filename}
```

Notes

- **Singleton:** Use `AssetConnector::getInstance($mysqli)`. Call `AssetConnector::reset()` if you need a fresh instance (rare).
- **Tags are merged:** If EXIF has keywords AND you pass `'tags' => 'custom'`, both are kept.
- **Title/description override:** If you pass these options, they REPLACE what EXIF extracted.
- **Collection auto-creation:** Collections are matched by `slug` (lowercased, hyphenated). "Company" → slug "company". If found, reuses existing. If not, creates new.
- **No double-uploads:** Files are sanitized with uniqid suffix, so same-name files never collide.
- **Storage quota:** Enforced by StorageService before S3 upload. If company is over quota, upload fails.