GLightbox Gallery Module

User Guide · Joomla Module · v4.14.12

A production-ready module that brings modern lightbox functionality to your website. Display images, videos, Google Maps, and custom HTML content in a beautiful, responsive lightbox — shown as a familiar thumbnail grid, a permanently-open inline slider, or a self-opening popup, whichever suits the page. Built on the GLightbox v3.3.1 JavaScript library and the Plyr v3.7.8 video player, with automatic thumbnail generation, an album mode, deep linking, a slideshow, a fullscreen button, EXIF photo details, tag filtering, structured data for search engines, and a settings preset system for saving and reusing your preferred look.

Gallery items are stored in their own database table rather than the module's settings field, so a gallery can hold thousands of photos without hitting Joomla's old size limit — and the items editor itself only ever draws the rows on screen, so opening, scrolling and editing a 2,000-item gallery in the admin feels exactly like editing one with ten. See §10.

Table of Contents

1. What Is This Module?
2. Requirements
3. Installation
4. Basic Setup
5. Display Modes: Gallery, Popup, Inline, Album
6. Content Types
7. Per-Item Status
8. Importing from a Folder
9. Bulk Actions
10. Fast Database Storage & the Items Editor
11. Settings Presets
12. Module Settings
13. Deep Linking to a Slide
14. Slideshow
15. Downloadable Files
16. Tag Filters and Pagination
17. Responsive Columns
18. Backdrop Colour
19. Caption Panel Customisation
20. Thumbnail Generation
21. Lightbox Images and WebP

22. Thumbnail Strip (Filmstrip)
23. Photo Information Panel (EXIF)
24. Where Generated Files Are Stored
25. Bulk-Generating and Maintaining Generated Files
26. Gallery Order
27. Structured Data (JSON-LD)
28. Popup Mode — Full Reference
29. Inline Mode — Full Reference
30. Custom CSS
31. Per-Item Options
32. Multi-Module Galleries
33. Caching
34. Practical Examples
35. Visual Customisation with CSS
36. Troubleshooting
37. Performance Tips
38. About GLightbox and Plyr

---

1. What Is This Module?

The GLightbox module integrates the GLightbox JavaScript library into Joomla as a ready-to-use gallery module. Without any coding, you can display images, embed YouTube and Vimeo videos, show Google Maps, and include inline HTML content in an elegant, animated lightbox overlay.

Beyond a simple lightbox, the module can now import straight from a media folder, generate correctly sized thumbnails automatically, group photos into folder-based albums, run itself as an unattended slideshow, popup, or permanently-open inline slider, show a visitor the camera and exposure details behind a photo, and describe the gallery to search engines. Items are kept in their own database table rather than the module's settings field, so even a gallery running into the thousands of photos stays quick to save, load and edit — see §10.

Typical use cases:

- Photography and portfolio galleries, optionally grouped into albums
- Product image showcases with video demos
- Team directories with photo and bio
- Location maps embedded in a lightbox
- Event galleries mixing photos and videos, run as a self-playing slideshow
- A landing page or hero section, shown inline and already open with no click needed — see §29
- Contact or newsletter forms shown in a lightbox popup, timed to page load or exit intent

- Large photography libraries where visitors filter by tag or browse by camera/exposure data

2. Requirements

Requirement	Version
Joomla	4.4 or newer (also runs on 5 and 6)
PHP	8.1 or newer
GLightbox library	v3.3.1 (included)
Plyr video player	v3.7.8 (included, used for direct MP4 playback)
GD extension	Recommended — needed for automatic thumbnail and lightbox-image generation. Without it the module still works, just without generated renditions.
EXIF extension	Optional — used to correct the orientation of phone photos, to read shooting details for the Photo Information Panel, and for "By Date Taken" ordering.
Items per gallery	1,000–2,000 comfortably per module instance, thanks to database-backed item storage — see §10.

No jQuery required. No external runtime dependencies — GLightbox and Plyr are bundled with the module, not loaded from a CDN.

3. Installation

1. Download the module package.
2. Log in to your Joomla backend.
3. Go to **Extensions** → **Manage** → **Install**.
4. On the **Upload Package File** tab, select the ZIP file and click **Upload & Install**.
5. Go to **System** → **Clear Cache**.

The module is now available at **Extensions** → **Modules** → **New** → **GLightbox Gallery**.

4. Basic Setup

1. Go to **Extensions** → **Modules** → **New** and select **GLightbox Gallery**.
2. Open the **Gallery Items** tab.

3. Either click **Add Item** to add slides one by one, or use **Import from Folder** at the top of the tab to create items from a media folder in one step — see §8.
4. Set the module **Position** and **Menu Assignment**, then save and publish.

5. Display Modes: Gallery, Popup, Inline, Album

Display Mode, at the top of the General tab, chooses what the module renders. **Gallery** is the default and is what every existing module stays on across an upgrade. The item list, the lightbox itself, the content types and the settings presets are identical underneath whichever mode is chosen — only how a visitor first encounters the gallery changes.

Mode	What it does
Gallery	The familiar grid of thumbnails, opening a lightbox on click.
Popup	No grid at all. The module opens a single item's lightbox by itself, on a schedule you control — see §28.
Inline	No grid and no overlay: a permanently-open slider rendered directly on the page, with its own arrows, thumbnail strip, counter and fullscreen button — see §29.
Album	Several folders in one module, each shown as one cover in the grid; opening a cover shows that folder's pictures — see below.

Album mode is really an organisation setting layered on top of Gallery mode rather than a fourth way of showing the gallery — see the Album mode section right below. It isn't available in Inline mode, which always shows a flat, single-level filmstrip.

Album mode

Switch **Album Mode** on under **Albums** on the Basic tab, then give each item an **Album** name. It is off by default, so upgrading leaves every existing instance exactly as it was.

Filling the albums. The fastest route is the import panel: put `{folder}` in its **Album** box and tick **With Subfolders** — `{folder}` is the file's own folder, so one import turns a whole tree into a set of albums. Importing one folder at a time in **Append** mode works just as well.

Which picture is the cover. Set **Album Cover** on the item you want. With no cover set, the first item in the album is used; with two marked, the first wins. A "Hidden from grid" item is never used as a cover.

Clicking a cover. Clicking a Cover decides what happens next:

- *The lightbox* (default) — goes straight into the album's pictures, full screen.

- *The album in the grid* — replaces the covers with that album's thumbnails and a way back above them; clicking one of those opens the lightbox, still inside the album.

In the sub-grid, tag filters and pagination act on the open album only, and the filter bar offers only the tags that album's pictures carry. A "Lightbox only" item inside an album stays out of the sub-grid but is still a reachable slide once the lightbox is open.

Captions. Two extra tokens: `{album}` (the album name) and `{count}` (how many pictures are behind the cover — resolves on a cover only). Put `{count}` on its own caption line so it disappears cleanly on non-cover items.

What follows from how it works. Every anchor carries a `data-gallery` value naming its album, so GLightbox narrows its own slide list to that album automatically. Three consequences: the slide counter, filmstrip and slideshow are album-relative; **Gallery Name** is ignored, because the album name is itself the grouping value; and **Item Order** applies inside an album only, with album order following first appearance in the item list. Items with no album stay visible and navigate among themselves as usual. Structured data becomes one `ImageGallery` per album.

6. Content Types

Images

Field	Description
Image	The full-size image shown in the lightbox.
Custom Thumbnail	Optional. Takes priority over an automatically generated thumbnail.
Title	Optional caption shown in the lightbox.
Description	Optional text shown near the title; accepts safe HTML and the photo-detail tokens.
Alt text	Alternative text (recommended for accessibility and SEO).
Status	Visible, Lightbox only, or Unpublished — see §7.

Automatic thumbnail generation only applies to local images; it never touches an image on another domain. SVG and AVIF sources are always shown but never re-rendered — see §20.

Videos — YouTube and Vimeo

Field	Description
Video URL	Full URL of the YouTube or Vimeo video.
Custom Thumbnail	Optional. If omitted, the platform's default thumbnail is used.

Supported formats: `https://www.youtube.com/watch?v=VIDEO_ID`, `https://youtu.be/VIDEO_ID`, `https://vimeo.com/VIDEO_ID`. A share link, a shortened link, and an embed link to the same video are normalised to the same result. YouTube videos are configured with privacy-enhanced mode, no related videos from other channels, and no annotations by default.

Autoplay is blocked by some mobile browsers — this is a browser-level restriction. Enabling **Mute Videos** (see §12) significantly improves the chance of autoplay working.

The provider (YouTube or Vimeo) is detected from the URL itself, not from a separate field — so a video item still plays correctly as an embed even if it's left on the generic **Video** type by mistake while its URL is actually a YouTube or Vimeo link. A URL containing neither `youtube.com`, `youtu.be`, nor `vimeo.com` is treated as a direct video file instead.

Direct Video (MP4)

Field	Description
Video URL	Full URL or path to the MP4 file, e.g. `/images/videos/demo.mp4`.
Custom Thumbnail	Required for MP4 files — there is no automatic thumbnail.

Direct video files play through the bundled Plyr player. Same-origin MP4 files are eligible for the download button — see §15. A value naming any other URL scheme (`javascript:`, `data:`, `file:`, …) is refused and the item is not rendered.

Google Maps / iFrame

The correct way to embed a map: open Google Maps, **Share** → **Embed a map**, copy the `src="..."` URL from the generated `<iframe>` code, and paste it into an **iFrame** item's URL field. The `https://` prefix is optional — the module adds it if missing. This item type also embeds any other external URL.

Inline HTML

Displays custom HTML — including embedded Joomla modules — inside the lightbox:

```
<div style="padding: 20px;">
  <h2>Contact Us</h2>
  <p>Email: info@example.com</p>
</div>
```

Embed a Joomla module by position or by ID:

```
{loadposition newsletter-form}
{loadmoduleid 42}
```

An unpublished Inline HTML item does not render its embedded module either.

7. Per-Item Status

Status	Behaviour
Visible	Default. Thumbnail shown in the grid, slide shown in the lightbox.
Lightbox only (no thumbnail)	No thumbnail in the grid, but the item stays reachable as a slide once the lightbox is open. Useful for large galleries where only a few cover images should show up front.
Unpublished	Not rendered at all — no thumbnail, no slide.

Hidden items carry a `glightbox-item-hidden` class for CSS targeting.

8. Importing from a Folder

The **Import from Folder** panel at the top of the Gallery Items tab creates one item per image, instead of adding slides one by one. It browses **one folder at a time**: a trail above the list goes back up, a filter box narrows the current level, and each folder shows its own image count (capped and shown as `500+` beyond that) and whether it has subfolders.

1. Upload images with Joomla's Media Manager as usual.
2. Open the import panel and pick a folder.
3. Choose **Import** (images directly in the open folder) or tick **With Subfolders** to walk the whole tree beneath it.
4. Choose whether to **Append** to the existing items or **Replace** all of them (replace asks for confirmation).
5. Set any defaults you want applied to every imported item, then run the import.
6. Save the module. Titles, alt text and descriptions can still be edited afterwards, item by item.

Recognised file types: JPG, JPEG, PNG, GIF, WebP, SVG, AVIF. Files are sorted naturally (`img2` before `img10`).

Import Defaults. The panel can pre-fill Title, Alt text, Description, Description Position, Effect, Status, Album, Download, Zoomable and Draggable on every item it creates. A default left on "Use default" is not written at all.

Wildcards in Title, Alt text and Description accept:

Wildcard	Meaning
`{filename}`	Tidied file name — `beach-sunset_02` becomes "Beach sunset 02"
`{rawname}`	Raw file name, without the extension
`{ext}`	File extension, lower-cased

Wildcard	Meaning
`{n}` / `{total}`	Running number within the import / total imported
`{folder}`	Name of the folder being imported — also used to fill **Album** for a folder-per-album import
`{title}`	Alt text and Description only — the title already resolved for that item

An unrecognised wildcard is left exactly as typed. The default Title pattern is `{filename}` and the default Alt text pattern is `{title}` — together they reproduce automatic titles from file names.

9. Bulk Actions

A checkbox column on the Gallery Items tab:

- **Select all / Select none** toggle every row; the header checkbox shows an indeterminate state on a partial selection.
- **Shift-click** selects a range, the same way it does in a file manager.
- **Delete selected** removes the chosen rows after confirmation.
- **Set Status** applies Visible / Lightbox only / Unpublished to every selected item at once; the selection is kept afterwards for a follow-up action.
- **Album** applies an album name to every selected item, with autocompletion offering the albums already in use.

Rows added later — manually or by a folder import — get their checkbox automatically.

10. Fast Database Storage & the Items Editor

Your gallery items live in their own database table, not in the module's settings. Older versions kept every item inside the module's `params` field, which Joomla caps at 65,535 bytes — with roughly 20 fields per item, that ceiling was reachable at somewhere between 40 and 120 items depending on how much text each one carried. Items now save to a dedicated table with no such limit, comfortably handling **1,000–2,000 items in a single module instance** while using only a few megabytes of database space.

Nothing to do — it upgrades itself. The first time you save an existing gallery after updating, its items are moved into the new table automatically; nothing changes in how the gallery looks or behaves, and older backups or exports remain compatible. New modules use the new storage from the moment you add their first item.

The items editor stays fast at any size. Because the admin field only ever builds the rows currently scrolled into view — plus a small buffer — instead of the whole list at once, opening a module with 2,000

photos in it feels the same as opening one with ten: scrolling is instant, and the page doesn't have to construct thousands of hidden form fields before it's ready to use.

- **Add Empty Item**, a button in the editor's own toolbar, starts a genuinely blank item — no type, no image, nothing assigned — and opens its panel immediately, for a slide whose shape isn't decided yet. Every other way an item enters the list — the media picker, a folder import — commits to at least a type and a source up front; this is the one path that doesn't.
- **Click a row** to open its edit panel directly beneath it, with all fields for that item's type — video items show the video URL field, inline items show the HTML editor button, and so on.
- **Drag the handle** on any row to reorder it; the list re-sorts as you drop it.
- **Select rows** individually, by shift-click range, or all at once — selection is tracked internally, so it survives scrolling even though the selected rows themselves may not be on screen.
- **Bulk actions** (status, album, tags, delete) apply to the current selection exactly as before — see §9.
- **Images and thumbnails** open Joomla's own native media picker.
- **Descriptions and inline HTML** open in the site's own configured editor (TinyMCE by default) via an **Edit HTML…** button next to the field.
- **Importing a folder is a single fast pass** — even a 2,000-image import completes in under a second, since there's no per-item DOM work to wait on. See §8.

Saving still goes through the module's own **Save** / **Save & Close** / **Apply** buttons as usual — there's nothing new to learn there.

11. Settings Presets

At the very top of the General tab, before any other setting, sits a collapsible **Presets** panel. Click the heading to open it. It offers three things, all of which fill in the form for you to review — nothing is saved until you press Joomla's own **Save** button.

Applying a built-in preset

Choose a preset from the dropdown and click **Apply**. The module's form fields update immediately; you can see exactly what changed before you commit.

The built-in library is organised into three groups:

Colour styles — change the lightbox backdrop and caption panel colours, filmstrip colours, and the slideshow progress bar. They leave layout and behaviour settings untouched, so you can layer a colour preset on top of your own column configuration without clobbering it.

Preset	Character
Dark	Near-opaque black, white strip highlights — the classic lightbox look.
Light	White with a frosted semi-transparent caption panel.

Preset	Character
Frosted glass	Dark navy with heavy blur and a blue accent, requiring a semi-transparent backdrop to be visible.
Cinematic black	Solid black, full opacity, red progress bar — cinema-style.
Warm sepia	Deep brown tones with a warm amber accent.

Layout — set Responsive Columns on and configure the column counts and thumbnail size for mobile, tablet, and desktop in one step.

Preset	Columns (mobile / tablet / desktop)	Ratio
Dense grid	3 / 4 / 6	1:1 square
Spacious	1 / 2 / 3	4:3 landscape
Uniform squares	2 / 3 / 4	1:1 square
Wide landscape	1 / 2 / 3	16:9 widescreen

Behaviour — change effects, filmstrip, slideshow, deep links and download in one step.

Preset	Effect
Minimal viewer	Fade effects, counter and filmstrip off, deep links and download off.
Full-featured	Counter and filmstrip on, EXIF panel on, deep links and download on.
Slideshow kiosk	Slideshow on at 4-second interval, loop on, filmstrip on.

> Presets compose: applying **Dark** and then **Dense grid** gives you both the dark look and the dense layout. One does not overwrite the other because each preset only touches its own group of settings.

Exporting your settings to a file

Export to file downloads a small JSON file containing a snapshot of every setting the panel can manage. It reads the live form, so it captures any edits you haven't yet saved. The file is named `glightbox-settings-YYYYMMDD.json`.

Use this to back up a carefully tuned configuration before experimenting, or to move identical settings to a second module or a different site.

Gallery items are **never included** in an export — only settings.

Loading settings from a file

Import from file reads a previously exported JSON file back into the form. Every recognised setting is applied to the corresponding form field; unrecognised keys (settings that no longer exist, or files from a different module) are silently skipped.

After importing, review the changes and press **Save** when you are happy with them. If anything looks wrong, press **Cancel** — nothing has been saved yet.

Saving your own presets

You can add your own presets to the built-in picker by editing a JSON file on your server. The file lives at:

```
/media/mod_glightbox/presets/custom.json
```

It is shipped empty (`[]`) and the module installer never overwrites it, so anything you add there survives every future update.

Open the file in a text editor (or the Joomla file manager) and add your presets inside the array. Each preset follows this shape:

```
[
  {
    "id": "my_brand",
    "group_key": "MOD_GLIGHTBOX_PRESET_GROUP_CUSTOM",
    "label_key": "my_brand",
    "params": {
      "overlay_color":    "#1a1040",
      "overlay_opacity":  "88",
      "progress_color":   "#c084fc",
      "filmstrip_color":  "#1a1040"
    }
  }
]
```

The fields:

Field	Purpose
`id`	A unique identifier — no spaces, letters and underscores only. Must not match any built-in preset id.
`group_key`	The optgroup this preset appears under. `MOD_GLIGHTBOX_PRESET_GROUP_CUSTOM` translates to "Custom presets" in English and "Saját sablonok" in Hungarian. You can invent your own group name — if it has no translation, the key itself is shown as the label.

Field	Purpose
`label_key`	The name shown in the dropdown. Same rule: translated if found, shown as-is otherwise. In practice, most custom presets use a plain descriptive string here that acts as its own label.
`params`	Any subset of settings from the whitelist (see below). Only keys listed here are written when the preset is applied — other settings are left untouched.

Finding the right parameter names and values. The easiest approach: configure the module exactly as you want, use **Export to file**, open the downloaded JSON, and copy the `params` block you want from it into your custom preset. The exported file uses exactly the same key names and value formats as `custom.json`.

Allowed parameter names (a selection of the most useful ones):

Category	Keys
Colours	`overlay_color`, `overlay_opacity`, `overlay_blur`, `overlay_saturate`, `desc_bg_color`, `desc_bg_opacity`, `desc_blur`, `desc_saturate`, `filmstrip_color`, `filmstrip_active_color`, `filmstrip_opacity`, `progress_track_color`, `progress_color`
Layout	`responsive_columns`, `columns_mobile`, `columns_tablet`, `columns_desktop`, `thumbnail_aspect_ratio_width`, `thumbnail_aspect_ratio_height`, `thumbnail_width`
Behaviour	`open_effect`, `close_effect`, `slide_effect`, `loop`, `show_counter`, `filmstrip`, `filmstrip_size`, `slideshow`, `slideshow_interval`, `deep_link`, `allow_download`, `exif_panel`
Caption	`desc_position`, `desc_align_h`, `desc_align_v`, `caption_position`, `caption_visibility`

Values are always strings, matching what the form field holds. Radio/toggle fields use `"0"` for off and `"1"` for on. Colour fields use `"#rrggbb"`. Opacity and numeric fields hold the plain number as a string: `"88"` not `88`. You can see the exact values any setting takes by using **Export to file** and looking at the resulting JSON.

Multiple custom presets — just add more objects inside the array, separated by commas:

```
[
  {
    "id": "brand_dark",
    "group_key": "MOD_GLIGHTBOX_PRESET_GROUP_CUSTOM",
    "label_key": "brand_dark",
    "params": { "overlay_color": "#1a1040", "overlay_opacity": "90" }
  },
  {
    "id": "brand_light",
    "group_key": "MOD_GLIGHTBOX_PRESET_GROUP_CUSTOM",
    "label_key": "brand_light",
    "params": { "overlay_color": "#f5f0ff", "overlay_opacity": "85" }
  }
]
```

The file must be valid JSON — a trailing comma after the last entry, or an unquoted value, will prevent all custom presets from loading. If your presets stop appearing, validate the file at jsonlint.com first.

12. Module Settings

Basic Settings

Setting	Default	Description
Display Mode	Gallery	Gallery, Popup, Inline, or Album — see §5.
Gallery Name	Empty	Group multiple modules into one gallery — see §32. Ignored in Popup and Album mode.
Layout	Default	Default (ul/li) or Minimal (anchor tags only).
Skin	Default	Clean or Default lightbox chrome.
Load CSS / Load JS	Yes / Yes	Disable if a template already includes GLightbox's assets, or to load them from elsewhere. JS loads once per page regardless of how many module instances are on it.

Thumbnail Options

Setting	Default	Description
Aspect Ratio (Width / Height)	16 / 9	The thumbnail crop ratio.

Setting	Default	Description
Thumbnail Width	200px	Accepts a bare number (read as pixels) or a CSS length; pixel widths snap to 200/400/600/800/1200.
Generate Thumbnails	Yes	See §20.
Thumbnail Folder	Empty (cache folder)	See §24.
Where to Store Them	Cache folder	Cache folder, or beside the images — see §24.

Lightbox Behaviour

Setting	Default	Description
Open / Close / Slide Effect	Zoom / Zoom / Slide	Zoom, Fade, Slide (slide-only), or None.
Zoomable	Yes	Allow zooming into images. Off automatically inside a popup.
Draggable	Yes	Allow drag/swipe navigation between slides.
Loop	No	Wrap from last item to first; also governs how the slideshow ends.
Show Slide Counter	Yes	"3 / 15" indicator, top-left.
Fullscreen Button	No	A button expanding the lightbox to fill the visitor's whole screen using the browser's own fullscreen mode. Off by default, so an upgrade adds no new button to an existing lightbox's row. Not supported on iPhone Safari.
Description Position	Bottom	Bottom, Top, Left, Right, or (in Popup mode) centred over the picture.
Bounce at the Ends	No	A small spring animation at the first/last item when Loop is off.

Other notable settings

- **Link to Single Slides** (default: No) — see §13.
- **Slideshow** and **Seconds per Slide** (default: No / 5s) — see §14.

- **Allow Download** (default: No) — see §15.
- **Backdrop Colour / Opacity** (default: empty = black at 92%) — see §18.
- **Structured Data (JSON-LD)** (default: No) — see §27.
- **Tag Filters** and **Images per Page** (default: off / 0) — see §16.
- **Responsive Columns** (default: off) — see §17.
- **Grid Caption** (default: empty) — a token-built caption under or over each thumbnail, using `{title}` `{description}` `{alt}` `{tags}` `{type}`.
- **Thumbnail Strip** (default: off) — see §22.
- **Photo Information Panel** (default: off) — see §23.
- **Generate Lightbox Images** and **Lightbox Maximum Width** (default: off) — see §21.
- **WebP Output (JPEG only)** (default: off) — see §21.
- **Autoplay Videos** / **Mute Videos** (default: Yes / Yes) — for the lightbox. Inline mode has its own separate pair of the same name — see §29.
- **Caching** / **Cache Time** (default: on / 900s) — see §33.
- **Custom CSS** (default: empty) — see §30.

13. Deep Linking to a Slide

Link to Single Slides puts the open slide in the page address (e.g. `yoursite.com/gallery#gb-portfolio-3`), so it updates as visitors browse and a specific slide can be linked and shared. A copy-link button appears next to the close button — the most reliable way to share a slide on a phone, where the address bar is hidden behind the lightbox. Turning this on also enables the copy-link button automatically. Off by default, since it visibly changes the address bar while browsing.

> **A deep link encodes a slide's position**, not a specific photo. If items are reordered or a new one is inserted near the front, an older link opens whatever now sits at that position. This is a natural consequence of how any position-based gallery link works, including merged multi-module galleries.

Suppressed entirely in Popup and Album mode (Album mode's deep links still open the lightbox directly, with the covers left on the grid behind it).

14. Slideshow

Adds a play button to the lightbox that advances the gallery on a timer. It never starts on its own — a visitor has to press play. A countdown bar next to the button shows the time left on the current slide; pause freezes it in place, and play continues from there rather than restarting.

Video items are allowed to play to the end instead of being cut off by the timer; a video that fails to autoplay (blocked by the browser) is skipped after one interval rather than stalling the slideshow. With Loop off, the last item gets its full turn and the lightbox then closes; with Loop on, it wraps around and keeps going.

Suppressed in Popup mode, and album-relative in Album mode.

15. Downloadable Files

Allow Download adds a download button for images and direct MP4 files. YouTube and Vimeo items never get one — offering to download them would breach their terms of service — and Inline HTML/iFrame items have no single source file, so they're excluded too.

The suggested file name comes from the item's title where set, falling back to the original file name. Same-origin files download directly; cross-domain files open in a new tab instead, since browsers ignore the `download` attribute across origins. Overridable per item — see §31.

16. Tag Filters and Pagination

Tag Filters shows a row of filter buttons above the gallery, built from a comma-separated **Tags** field on each item. Selecting a tag narrows both the grid and the lightbox together; the lightbox only ever navigates the currently filtered set. Works in both layouts.

Images per Page turns on numbered grid pagination. The lightbox still navigates every filtered item regardless of which page is showing — pagination is a grid-display convenience, not a lightbox restriction. Set to 0 (default) to keep the grid unpaginated.

In Album mode, both act on the open album's sub-grid, and the filter bar only ever offers tags actually present in that album.

Both are suppressed in Popup mode.

17. Responsive Columns

Responsive Columns switches the Default layout from a fitted flexbox grid to a fixed number of columns per breakpoint — mobile, tablet, desktop — instead of thumbnails that size themselves to fit. Off (the default) keeps the existing flexbox behaviour unchanged, so upgrading changes nothing until it's switched on. Default layout only.

18. Backdrop Colour

Backdrop Colour and **Backdrop Opacity** set the colour behind the lightbox. Opacity accepts either a percentage (0–100) or a CSS alpha value (0–1) — `78` and `0.78` mean the same thing. Leave both empty to keep GLightbox's own default (solid black at 92%). If two merged galleries on the same page set this differently, whichever renders first on the page wins, since only one backdrop is ever shown at once.

19. Caption Panel Customisation

Caption Background Colour and **Caption Background Opacity** set the surface the caption sits on. Beside or below the image, the caption is a flat panel and the colour paints it directly; on a phone the caption lies over the photo, so instead of a flat band the colour rebuilds GLightbox's fade — transparent at the top, solid at the foot — scaling both ends together with the opacity.

Caption Alignment Beside the Picture and **Caption Alignment Above or Below** position the text inside its panel; only one applies to a given caption, depending on which side it's on. An item overriding its own position with `data-desc-position` still gets the matching alignment.

Caption Backdrop Blur (0–40px) and **Caption Backdrop Saturation** (100% = unchanged) add a frosted-glass effect behind a semi-transparent caption panel — useful mainly for the phone overlay, since beside/below the image the caption doesn't sit over the photo. **Overlay Backdrop Blur** and **Overlay Backdrop Saturation** do the same to the lightbox backdrop itself (lower **Backdrop Opacity** too, or there's nothing to see through). This is the most expensive setting in the module — the browser re-blurs the whole page on every frame, slide animations included.

In Popup mode, **Description Position** offers a fifth value: the caption centred over the picture, cropped to fill an 80vh box (60vh on narrow screens), sitting over a scrim with the pointer passing through to the picture beneath. It's an ordinary entry in the position list, so it also works gallery-wide or per item outside a popup. Videos and iframes fall back to Bottom, as they already do for Left/Top.

20. Thumbnail Generation

With **Generate Thumbnails** on (the default), every local image gets a thumbnail cropped to the configured aspect ratio, at the size the grid actually shows, at 1x and 2x for high-density screens. A 24-photo gallery of 4000×3000 phone pictures drops from roughly 100 MB of downloads to well under a megabyte.

- **Refresh.** The cache key includes the source file's modification time, so replacing an image regenerates its thumbnail on the next page view; the old file is deleted at the same moment.
- **Requirements.** The GD extension. Rotation of phone photos additionally uses the EXIF extension.
- **What falls back to the full image, with no error:** images on other domains, sources smaller than the thumbnail itself, and images above roughly 50 megapixels. An item with a hand-picked Custom Thumbnail always keeps it.
- **SVG and AVIF are shown but never re-rendered.** Joomla's image class only reads GIF, JPEG, PNG and WebP, so neither format can be a generation source — for AVIF this holds even where GD itself supports it. Both are still offered by the folder import and served at their own size.
- **Which formats can be generated from depends on the GD build.** GD is often compiled without WebP support on shared hosting; the module checks and treats an unsupported source as un-processable immediately, rather than trying and failing on every page view.
- **Non-pixel widths** (`25%`, `15em`) generate at 400px, since the actual rendered size depends on the visitor's viewport.
- Animated GIFs are reduced to their first frame — a GD limitation.

21. Lightbox Images and WebP

Generate Lightbox Images (default: off) extends the same idea to the slide the visitor opens, which by default is still the untouched original. When on, the lightbox is offered a ladder of resized copies (640, 1280, 1920, 2560px, capped by **Lightbox Maximum Width**), and the browser picks the smallest one that fills its viewport — a phone opening a 4MB, 4000px photo gets a few hundred kilobytes instead. The original stays the link target for the download button and "open in new tab," and remains the widest rung for large displays. A rung within 90% of the source width is skipped, so small images get no ladder and are served whole.

WebP Output (JPEG only) (default: off) additionally writes a WebP copy of each generated image alongside the original-format one — typically about a third smaller at the same visible quality. The grid offers both through a `<picture>` element and the browser chooses; the lightbox decision happens on the visitor's machine, so a page cache or CDN never serves one visitor's choice to another. PNG, GIF and already-WebP sources are left untouched (lossy WebP damages flat graphics, and converting a GIF would drop its animation).

How a large gallery fills in. Renditions are generated on the page views that need them, and generation is capped per request (a time budget between 3–15 seconds, and a 200-file ceiling) so a large gallery can never time out a page load. What a request couldn't reach is served as the full image for that one render only; the next visit finds the files already on disk. A page view that ran short is not cached, so the gallery keeps filling in until it's warm — visiting the page a few times ahead of a launch brings it fully up to date immediately. See also §25 for generating everything up front instead of waiting on visitors.

22. Thumbnail Strip (Filmstrip)

Thumbnail Strip puts a scrollable row of thumbnails under the lightbox image. It opens with the gallery; a toolbar button closes and reopens it. The current slide stays centred as you browse, and clicking a thumbnail jumps to it. There's no scrollbar — drag with the mouse, or use touch/trackpad as normal.

Strip Thumbnail Size (40–200px) sets the tile size; **Strip Background Colour**, **Strip Background Opacity** and **Active Thumbnail Colour** style the bar. It reuses the grid's own generated thumbnails, adding no extra downloads, and follows the same filtered slide set the lightbox does — pagination doesn't affect it, since the lightbox always navigates the full filtered gallery.

Opening the strip shortens the image rather than covering it. Hidden below 769px and while an image is zoomed — zooming stops a running slideshow, and pressing play while zoomed in zooms back out first.

Suppressed in Popup mode.

23. Photo Information Panel (EXIF)

Photo Information Panel adds a button to the lightbox opening a small panel with shooting details read straight from the photo: camera, lens, focal length, aperture, shutter speed, ISO, and the date taken. Nothing needs entering by hand. The button appears only where there's something to show — a screenshot, PNG, video, or a photo an editor stripped of metadata simply gets no button.

Location data is never read or shown. Phone cameras record GPS coordinates to within a few metres; a public gallery has no business publishing where people live, so the module doesn't read that section of the file at all.

Fields to Show picks which of the seven details appear, always in the module's own order regardless of tick order; leaving everything unticked shows all of them.

The same values work as tokens — `{camera}`, `{lens}`, `{focal}`, `{aperture}`, `{shutter}`, `{iso}`, `{taken}` — in both the caption and an item's own Description field, whether or not the panel itself is switched on. A caption line whose photo tokens are all empty is dropped, so photographs and scans can sit in the same gallery without per-item fiddling. The folder import lists the same tokens beside its Description box, so a whole folder can be imported with them already in place.

Requires PHP's `exif` extension. Without it the panel stays hidden and the tokens resolve to nothing.

24. Where Generated Files Are Stored

Where to Store Them applies to both thumbnails and lightbox renditions.

Cache folder (default) — everything in one shared location, `media/cache/mod_glightbox/` by default or wherever **Thumbnail Folder** points. It sits outside the Media Manager root, so generated files never appear in an image picker, and uninstalling the module removes them.

Beside the images — puts renditions in a `thumbs/` subfolder next to each source photo (`images/gallery/sunset.jpg` → `images/gallery/thumbs/<hash>_400x225_c_<time>.jpg`), matching Joomla's own convention. They travel with the originals through backups and folder moves. Two things to know: they **do** show up in image pickers (the module's own folder import skips them), and uninstalling the module does **not** remove them, since there's no central list of where they ended up — deleting a `thumbs/` folder by hand is always safe, since the gallery simply regenerates what it needs.

Whichever option is chosen, only files matching the module's own generated naming scheme are ever written or deleted — nothing else in those folders is touched.

25. Bulk-Generating and Maintaining Generated Files

Generate Images, in the Files settings group, makes thumbnail and lightbox renditions immediately instead of waiting for visitors to trigger them one page view at a time.

- **Generate what is missing** — skips anything already on disk. Run after adding new photographs.
- **Regenerate everything** — rewrites all of it, which is what a changed aspect ratio or thumbnail width needs.

One image per request, so it can never time out however large the gallery is, with a progress bar and a Stop button that takes effect between images. It uses the settings as they currently stand on screen, so it can be run before saving — but files made from unsaved settings are only picked up by the live page once those settings are actually saved. The natural order is: change the settings, generate, then save.

Generated Files reports total disk usage across the whole site and offers two cleanup actions:

- **Remove unused** — deletes only files whose source image is no longer referenced by any gallery on the site (published or not).
- **Remove all** — deletes every generated file, in use or not; the only fix for a setting change (like a new aspect ratio) that leaves the old files looking current. The first visit to a large gallery afterwards is slower while everything rebuilds.

Both actions require back-end edit access and are sent as POST requests, so a link checker or crawler cannot trigger one by accident.

26. Gallery Order

Gallery Order, on the Gallery Items tab, decides item order. **Manual** (the listed order) is the default and what every existing gallery already does.

Option	Behaviour
By Title / By File Name	Natural, case-insensitive comparison (`photo 2` before `photo 10`); accent-aware where the `intl` extension is available.
By Date Taken	Reads the EXIF capture date, switching EXIF reading on automatically for this purpose (independent of the info panel setting).
Random	Reshuffles on every visit — applied after the module cache, so a shuffle is never frozen for the length of Cache Time.

Items with nothing to sort on (no title, no capture date) always sort last, in either direction. Items comparing equal keep their editor order.

27. Structured Data (JSON-LD)

Structured Data describes the gallery to search engines as a schema.org `ImageGallery`, using each item's title and description. Only image items are included — video items are left out, since a proper `VideoObject` needs fields the module doesn't collect, and Inline HTML/iFrame items aren't media. A gallery with no qualifying images produces no structured data block at all. In Album mode, each album becomes its own `ImageGallery`, named after its cover title or album name. Suppressed in Popup mode.

28. Popup Mode — Full Reference

A popup renders no grid. It emits the first item of the Gallery Items list (after ordering — **Gallery Order** picks it; Random rotates the popup on its own) as a single hidden trigger, and opens it itself when the page loads. Because it's an ordinary gallery slide, every item type works, including Inline HTML carrying

`{loadposition}` or `{loadmoduleid}` — a way to put an entire Joomla module inside a popup. Size, effects, backdrop and captions all come from the usual General tab settings.

A popup is a module like any other, so its own menu assignment, access level and publishing dates decide where and when it appears — there's no second targeting system to keep in sync.

The Popup tab

Setting	Description
Open After	Seconds to wait after the page loads. 0 opens it immediately.
Close After	Seconds before it closes itself. 0 leaves closing to the visitor.
Show Countdown	A bar along the top edge counting down to the automatic close. Needs Close After.
Show	Every page load, once per browser session, or once per period.
Period / Period Unit	Minutes, hours, or days.
Remember In	Local storage or a cookie.
Open On	Page load, or exit intent.
Close When The Video Ends	For a video item, close as the clip finishes.
Offer A Way To Opt Out	A tick box that stops the popup coming back for that visitor.
Close Button / Click Outside / Escape Key	Each switchable individually; switching off all three brings the close button back automatically — a popup a visitor cannot dismiss is not something the module will render.
Swipe To Close	On by default; also governs the sideways drag animation on a single slide.

Exit intent. With Open On set to exit intent, the popup arms a detector for the pointer crossing the top edge of the window (toward the address bar/tabs) rather than opening right away; Open After becomes the arming delay. It opens once per arming, and only where `(pointer: fine)` matches — so it never fires on a phone or tablet. Use page load if it needs to reach every visitor.

Videos. Close When The Video Ends closes as the clip finishes. Close After stays armed as a fallback for the one case that would otherwise leave the popup open indefinitely: an unmuted video the browser refuses to autoplay. Once playback genuinely starts, the fallback is cancelled and the countdown bar comes down with it.

Frequency. Worked out entirely in the browser, never on the server — the module's cached output is identical for every visitor, so the server has no way to know who's already seen a given popup. The "seen it" record is written when the popup **opens**, not when it's dismissed, so reloading right after closing it still counts as having seen it. Local storage never travels with a request; a cookie does, but survives where local storage is blocked — either way, a visitor blocking all storage sees the popup every time rather than never. Cookies are written with `path=/`, `SameSite=Lax`, and `Secure` on an HTTPS site (omitted on plain HTTP, since a browser would otherwise discard a Secure cookie outright and the popup would never remember anything).

Opt-out. Ticking the box records the decision under its own key, checked before the frequency rule — an opt-out holds even on "every page load." It only takes effect while the setting itself is switched on.

What popup mode switches off. Deep links, the slideshow, JSON-LD, tag filters, the filmstrip, pagination, the end bounce, and responsive columns — each describes a grid or a set of slides, and a popup is one slide opened by the page. Grid thumbnails aren't generated either, since there's no grid. **Generate Lightbox Images** still applies, since that's the file the open slide loads. Zoom is switched off, whatever the item or gallery setting says (a zoomed slide hides its caption in GLightbox, which would strand a popup's message). **Gallery Name** is ignored, since a shared name would hand the popup's slide to every other gallery carrying it and pull theirs into the popup too.

29. Inline Mode — Full Reference

Inline mode renders a permanently-open slider — stage, arrows, thumbnail strip, slide counter and title — directly on the page, in normal document flow. Nothing is drawn as an overlay unless **Open Full-Screen Lightbox** is switched on; with it off, the instance never even asks the browser to load the vendored GLightbox library, its stylesheet, or the shared init script — a genuinely standalone slider. Every content type works: images, YouTube and Vimeo, direct video files, and inline HTML.

The Inline tab

Setting	Default	Description
Stage Aspect Ratio Width / Height	16 / 9	The stage's own aspect ratio, independent of the grid's Aspect Ratio setting.
Background Colour	Empty (dark)	Behind letterboxed images that don't fill the stage.

Setting	Default	Description
Blurred Image Backdrop	No	Fills the letterboxed area with a blurred, enlarged copy of the current picture instead of a flat colour.
Prev/Next Arrows	Yes	Show or hide the corner arrows.
Counter	Yes	The "3 / 12" slide counter, top-left, next to the title.
Thumbnail Filmstrip	Yes	Whether the filmstrip — and the visitor-facing show/hide button for it — exists at all. Reuses the same **Filmstrip** group's size/colour settings (Module tab) the real lightbox's own filmstrip uses.
Open Full-Screen Lightbox	Yes	See above — the one setting that decides whether this instance depends on the lightbox library at all.
Fullscreen Button	Yes	A separate corner button that expands the stage to fill the screen via the browser's own fullscreen mode, independent of Open Full-Screen Lightbox. Not supported on iPhone Safari — a platform limitation, not something this module can work around; the button still shows there, it just has nothing to do when pressed.
Autoplay Videos	Yes	For video slides on the stage. Independent of the lightbox's own Autoplay Videos setting (Module tab) — Inline mode can behave differently.
Mute Videos	Yes	For video slides on the stage. Independent of the lightbox's own Mute Videos setting.
Touch Swipe	Yes	Left/right swipe navigation.
Keyboard Navigation	Yes	Arrow-key stepping.
Loop	No	Whether arrows, filmstrip, swipe and keyboard navigation wrap around at the ends. Autoplay follows this too: on, it wraps continuously; off, it stops at the last slide instead of jumping back to the first.
Starting Item	1	Which item (counting from 1) the stage opens on. Out of range, it falls back to the first item.

Setting	Default	Description
Autoplay	No	The stage advances on its own, with a play/pause button in the corner and a pausable countdown bar along the top edge. Manual navigation pauses it.
Start Automatically	No	Needs Autoplay on. Off, the visitor has to press play once; on, the slideshow starts the moment the stage appears.

Zoom and drag. The per-item and gallery-wide Zoom/Drag settings that apply to the real lightbox don't have a separate inline equivalent — they govern the lightbox that Open Full-Screen Lightbox may open, not the stage itself.

What doesn't apply here. Deep links, tag filters, pagination, responsive columns, and Structured Data (JSON-LD) each describe a grid or a paged set of slides; Inline mode is one continuously-open slider, so none of them have anything to attach to. Album mode isn't supported either — an inline instance always shows a flat, single-level filmstrip.

30. Custom CSS

The Advanced tab has a **Custom CSS** field, right after Module Class Suffix — a plain textarea for CSS the site owner wants to write directly rather than ask for a setting.

It's added to the page head **after every other stylesheet the module registers** — the grid, the lightbox chrome, the caption panel, the download button, all of it — so a plain selector overrides the module's own output with no `!important` needed:

```
.mod-glightbox .glightbox-thumbnail {
    border-radius: 0;
}
```

Empty by default, so upgrading changes nothing on an existing gallery. It applies whether or not **Load CSS** is switched on, since it's the site owner's own rules rather than part of the vendored stylesheet. The field is emitted as-is, with no validation — a typo here is a typo on the live page.

31. Per-Item Options

Each gallery item can override the module's general settings:

Option	Description
Title / Description / Alt text	Item-specific text.
Custom Thumbnail	Overrides an auto-generated thumbnail.

Option	Description
Width / Height	Overrides the default lightbox dimensions for this item.
Description Position	Per-item placement override.
Effect	Per-item slide animation override.
Zoom / Drag	Enable or disable per item.
Download	Enable or disable the download button per item (images and MP4 only).
Status	Visible, Lightbox only, or Unpublished.
Album	Which album (if any) this item belongs to.
Tags	Comma-separated tags for the tag filter.
Album Cover	Marks this item as its album's cover.

32. Multi-Module Galleries

Give two or more module instances the same **Gallery Name** to merge them into one gallery, navigable across all of them together — across both layouts, with no limit on the number of modules.

Requirements: the Gallery Name must match exactly (case-sensitive); all modules must be published and assigned to the same page; clear the Joomla cache after any change.

Where a per-gallery setting can only apply once — **Backdrop Colour/Opacity**, **Link to Single Slides** — whichever module renders first on the page wins.

Gallery Name is ignored in both Popup mode (it would hand the popup's slide to every gallery sharing the name) and Album mode (the album name is itself the grouping value).

33. Caching

Caching and **Cache Time** (Advanced tab, default: on / 900 seconds) control whether the module keeps its prepared gallery output between page views. Preparing a gallery is the expensive part — measuring every source image, reading EXIF headers, checking which generated renditions exist — so a cached view skips all of it.

Three conditions must all be true before anything is cached:

- **Caching must be on in Global Configuration.** The module's own setting is a veto, not an override — it can turn caching off for this one gallery, but never on for a site that has it off, the same rule that applies to every Joomla module.

- **The visitor must be a guest.** Joomla disables module caching for logged-in users, so an editor always sees the live gallery.
- **No item may embed another module.** A `{loadposition}` / `{loadmoduleid}` item's output can depend on who's viewing it, so a gallery containing one always renders uncached.

The cache needs no manual clearing — its key is built from the module's own settings and item list, so saving any change lands on a new key and the next view rebuilds automatically. Upgrading the module does the same.

The one case that genuinely waits for **Cache Time** to expire is replacing an image file on disk under its old name without touching the module itself — the gallery keeps serving the previous renditions until the cache entry expires. If that's a routine part of how a site is managed, set Caching to **No caching** on that module.

34. Practical Examples

Product Showcase — Layout: Default · Gallery Name: `products`

- Product 1: Image with price in description
- Product 2: Image with "Buy now" link in description
- Product 3: YouTube demo video
- Product 4: Inline HTML with a specifications table

Team Directory — Layout: Default

- Photo + bio in description; one member's item is Inline HTML with a contact form; another is a Vimeo introduction video.

Event Gallery, Run as a Slideshow — Gallery Name: `event2025` · Slideshow: On, 6s · Loop: On

- Welcome speech (YouTube), keynote (Vimeo), 20 event photos with automatic thumbnails, a venue map (iFrame), and a feedback form (Inline HTML). Placed on a lobby screen with Loop on, this runs unattended once someone presses play — photos and videos in sequence, each video playing to completion.

Photography Library as Folders of Albums — Album Mode: On · Clicking a Cover: The album in the grid

- One folder-per-shoot import with `{folder}` in the Album wildcard and **With Subfolders** ticked turns an entire photo library into a wall of covers a visitor steps into, each with its own tag filter and slide counter.

Welcome Popup — Display Mode: Popup · Open On: Page load · Open After: 2s · Show: Once per session · Offer a way to opt out: On

- A single Inline HTML item embedding a `{loadposition}` newsletter-signup module, closing on Escape/click-outside/close-button as usual.

Multi-Language Site with a Multilang Article Module. GLightbox reinitialises automatically via `MutationObserver` when content is swapped in dynamically, so galleries embedded in language-switched

content work without extra configuration.

35. Visual Customisation with CSS

HTML structure — Default layout

```
<div id="mod-glightbox-{ID}" class="mod-glightbox">
  <ul class="glightbox-gallery">
    <!-- Image item, with a generated thumbnail -->
    <li class="glightbox-item">
      <a href="large.jpg" class="glightbox glightbox-thumbnail">
        <img src="thumb.jpg" srcset="thumb.jpg 1x, thumb@2x.jpg 2x" alt="…"
loading="lazy">
      </a>
    </li>
    <!-- Video item (YouTube / Vimeo / MP4) -->
    <li class="glightbox-item">
      <a href="https://youtube.com/…" class="glightbox glightbox-thumbnail glightbox-
video-play">
        <img src="thumb.jpg" alt="…" loading="lazy">
      </a>
    </li>
    <!-- Inline HTML item -->
    <li class="glightbox-item">
      <a href="#inline-{ID}-2" class="glightbox glightbox-thumbnail">
        <img src="thumb.jpg" alt="…" loading="lazy">
      </a>
      <div id="inline-{ID}-2" style="display:none;" class="glightbox-inline-content"><!--
content --></div>
    </li>
    <!-- "Lightbox only" item: no thumbnail box, still a slide -->
    <li class="glightbox-item glightbox-item-hidden" style="display:none">
      <a href="large.jpg" class="glightbox glightbox-thumbnail">
        <img src="thumb.jpg" alt="…" loading="lazy">
      </a>
    </li>
  </ul>
</div>
```

A wrapper showing exactly one visible tile (a genuine one-item gallery, or every other item set to "Lightbox only") additionally receives `mod-glightbox-single`:

```
<div id="mod-glightbox-124" class="mod-glightbox mod-glightbox-single">
```

Useful for a single hero image with a gallery behind it — `.mod-glightbox-single .glightbox-thumbnail { width: 100%; }`. Not applied in Popup mode. The tile also gets a small photo-count strip along its bottom edge and, on desktop hover/focus, a circular play-style icon — both automatic, nothing to switch on separately.

HTML structure — Minimal layout

```
<a href="large.jpg" class="glightbox glightbox-thumbnail">
  <img src="thumb.jpg" srcset="thumb.jpg 1x, thumb@2x.jpg 2x" alt="…" loading="lazy">
</a>
```

No wrapper `div`, `ul`, or `li` — ideal for fully custom layouts.

HTML structure — Popup layout

```
<a href="…" class="glightbox mod-glightbox-popup-source glightbox-mod-7"
   style="display:none" aria-hidden="true" tabindex="-1"></a>
```

One hidden trigger, no grid. Used only for Popup mode.

Class reference

Class	Element	Description
`.mod-glightbox`	wrapper	Default layout only. Also receives any Module Class Suffix.
`.mod-glightbox-single`	wrapper	Present when the grid shows exactly one tile — see above.
`.glightbox-gallery`	`<ul>`	The thumbnail list, Default layout only.
`.glightbox-item`	`<li>`	One list item per gallery entry, Default layout only.
`.glightbox-item-hidden`	`<li>` / `<a>`	"Lightbox only" items, alongside `display:none`.
`.glightbox`	`<a>`	Every lightbox trigger link — required by GLightbox itself.
`.glightbox-thumbnail`	`<a>`	Every trigger link, used for thumbnail sizing and hover effects.
`.glightbox-video-play`	`<a>`	Video items — renders the play button overlay via `::before` / `::after`.
`.glightbox-inline-content`	`<div>`	Hidden container for Inline HTML items.
`.glightbox-slide-counter`	`<div>`	The "3 / 15" overlay injected into `<body>` when the lightbox is open.
`.mod-glightbox-download`	`<button>`	The lightbox's download button, when Allow Download applies.

Class	Element	Description
`.mod-glightbox-permalink`	`<button>`	The copy-link button, when Link to Single Slides is on.
`.mod-glightbox-slideshow`	`<button>`	The slideshow play/pause button.
`.mod-glightbox-progress`	`<div>`	The slideshow countdown bar; carries `.is-visible` while active or paused.
`.mod-glightbox-popup-source`	`<a>`	The hidden trigger in Popup mode.

Thumbnail dimensions are set by inline styles based on **Thumbnail Width** and **Aspect Ratio**. Override from your own stylesheet, or from the module's own **Custom CSS** field (see §30), which is guaranteed to load last:

```
.mod-glightbox .glightbox-thumbnail {
    width: 180px;
    aspect-ratio: 4 / 3;
}
```

Common customisations:

```
/* Remove the hover lift effect */
.mod-glightbox .glightbox-thumbnail:hover {
    transform: none;
    box-shadow: 0 2px 8px rgba(0, 0, 0, 0.1);
}

/* Add a border to thumbnails */
.mod-glightbox .glightbox-thumbnail {
    border: 2px solid #dee2e6;
    border-radius: 8px;
}

/* Style the slide counter */
.glightbox-slide-counter {
    font-size: 13px !important;
    background: rgba(0, 0, 0, 0.6) !important;
    border-radius: 20px !important;
    padding: 4px 14px !important;
}

/* Target a specific module instance by ID */
#mod-glightbox-42 .glightbox-thumbnail {
    border-radius: 50%; /* circular thumbnails for this module only */
}
```

> Use your browser's developer tools (F12 → Inspector) to confirm class names and IDs on the live page before writing custom CSS — this module writes several context-dependent classes (album, popup, filmstrip, caption panel) not all listed above.

36. Troubleshooting

Gallery not showing

1. Confirm the module is published and assigned to the current page.
2. Go to **System** → **Clear Cache**.
3. Verify the module position exists in your template.
4. Check the browser console (F12) for JavaScript errors.

Images not loading

1. Verify image paths and file permissions (644).
2. Confirm the format is supported: JPG, PNG, GIF, WebP, SVG, AVIF.

Thumbnails (or lightbox renditions) not being generated

1. Confirm the GD extension is enabled (**System** → **System Information** → **PHP Information**).
2. Confirm **Generate Thumbnails** (or **Generate Lightbox Images**) is on.
3. Check that the thumbnail folder is writable — by default `media/cache/mod_glightbox/`, or wherever **Thumbnail Folder** points.
4. SVG images, images on another domain, sources smaller than the target, and unusually large sources are served in full by design — not a fault.
5. If a replaced image under the same file name still shows the old thumbnail, clear the Joomla cache and hard-reload (Ctrl+F5 / Cmd+Shift+R) — the module keys renditions on the file's modification time.
6. On a large, freshly imported gallery, some images may still be waiting their turn — see §21 on how filling in works, or use **Generate Images** in the Files settings to force it immediately.

Videos not playing

1. Verify the YouTube/Vimeo URL and that the video is public.
2. For MP4, confirm the file path is correct and accessible.
3. Enable **Mute Videos** if relying on autoplay.
4. Clear the browser cache and retry.

Google Maps not displaying

1. Use the embed URL from **Share** → **Embed a map**, not the regular sharing link.
2. Confirm the item type is iFrame, not Image or Inline HTML.
3. Test the URL directly in a new browser tab.

Multiple modules not merging

1. Verify the Gallery Name is identical, spelling and capitalisation included.
2. Confirm all modules are published and assigned to the same page.
3. Clear the Joomla cache after any change.

Deep link opens the wrong slide. A deep link encodes a slide's position, not a specific photo — reordering or inserting items changes what an old link points to. Re-share the link from the current gallery to fix it.

Popup not appearing

1. Confirm **Display Mode** is set to Popup and the module is published/assigned correctly.
2. Check the frequency setting (**Show**) — "Once per session/period" means it will not reappear until that period elapses; clear the relevant cookie or local storage entry to force it again while testing.
3. If **Open On** is Exit Intent, remember it only arms on devices with a precise pointer (desktop mouse), never on touch-only devices.
4. Confirm the opt-out box hasn't been ticked during earlier testing.

Album not grouping correctly. Confirm every intended item has the exact same **Album** value (case-sensitive) and that **Album Mode** is switched on. Use the bulk **Album** action with its autocompletion to avoid typos splitting one album into two.

Custom CSS not applying. Confirm the CSS was saved. Since Custom CSS loads last and needs no `!important`, an override that still isn't working usually means the selector itself doesn't match the rendered markup — check the class reference above or inspect the element directly.

37. Performance Tips

- Leave **Generate Thumbnails** on — it typically cuts what a gallery grid downloads by well over 90%.
- Consider **Generate Lightbox Images** and **WebP Output** for large source photos, especially on a gallery aimed mostly at mobile visitors.
- Use **Generate Images** in the Files settings ahead of a launch, so the gallery is fully warm on the first real visit instead of filling in over several page views.
- Enable lazy loading where your template supports it for image-heavy pages.
- Optimise image files before uploading.
- Use Custom Thumbnails only where a specific crop is needed — let automatic generation handle the rest.
- There's no need to split a large gallery across modules just to keep the admin usable — database-backed storage and the windowed items editor keep a single module comfortable up to 1,000–2,000 items (see §10). Splitting across modules or Album Mode is still worthwhile for how visitors browse the gallery, just not required for admin performance.
- Leave **Caching** on, and set an appropriate **Cache Time** for how often your images actually change.

38. About GLightbox and Plyr

This module is built on the open-source GLightbox library by biati-digital, and uses the open-source Plyr video player for direct video playback — both released under the MIT License.

Component	Version	License
GLightbox	3.3.1	MIT
Plyr	3.7.8	MIT