Form Expiry Field for RSForm! Pro — Documentation

Version: 1.9.0
Compatibility: Joomla 4.x / 5.x + RSForm! Pro 3.x+ (scheduled notifications: Joomla 4.1+)
Developer: László Blahó — extensions.blaho.hu
Email: extension@blaho.hu
License: GNU General Public License v2.0+
Last updated: 2026-11-11

Table of Contents

1. Description
2. Installation
3. File Structure
4. Adding and Configuring the Field
5. Properties Reference
6. Timezone Modes
7. Countdown Timer
8. Capacity Limit
9. Admin Email Notifications
10. CSS Customization
11. Frontend Output
12. Admin Preview
13. Troubleshooting
14. Updating
15. Status Dashboard
16. Scheduled Notifications
17. Security Notes

---

1. Description

The Form Expiry Field plugin adds a custom field type to RSForm! Pro that controls when a form accepts submissions — an opening time, a closing time and a maximum number of submissions.

Main features

- **Opening date (1.7.0)** — The form is "not open yet" until a set time, then opens in place without a page reload
- **Expiry date** — Exact closing time (YYYY-MM-DD HH:MM)
- **Capacity limit** — Maximum number of submissions, accurate even when many people submit at once (1.7.0), optionally counting only submissions since the opening date
- **Capacity integrity (1.9.0)** — Optionally only confirmed submissions count; one submission per email address (or any other field value) and per logged-in user
- **Waiting list (1.9.0)** — After the maximum is reached, a configurable number of further submissions is accepted; applicants learn by email whether they have a place or are on the waiting list
- **Submit blocking** — On not-yet-open, expired, full or closed forms the submit buttons are disabled automatically, and the server rejects the submission even with JavaScript switched off
- **Notification popup** — Optional modal dialog, usable with keyboard and screen readers
- **Form overlay** — Striped overlay on a form that cannot be submitted
- **Warning period** — Configurable advance warning in days/hours, with an automatic "This form will close in X days!" message
- **Countdown timer** — Formatted static date or live countdown (to the opening before it, to the expiry after it); every state change happens in place, without a page reload
- **Server-synchronised clock (1.7.0)** — A visitor's wrong computer clock cannot open the form early or keep it open late
- **Timezone handling** — 3 modes: server, visitor's browser, fixed timezone
- **Email notification** — Admin email when the form expires or the capacity is reached, once per event; with a scheduled task even if nobody opens the form (1.8.0)
- **Status dashboard (1.8.0)** — Administrator module on the Home Dashboard: the state, next opening/closing and capacity of every form in one place; from 1.9.0 with Close / Reopen buttons and the number of blocked submission attempts
- **Dynamic closure** — Forms can be closed and reopened from the dashboard (1.9.0) or from code (globally, for every visitor)
- **Multilingual** — English and Hungarian language files

---

2. Installation

Requirements

- Joomla 4.0 or newer
- RSForm! Pro 3.0 or newer
- PHP 7.4 or newer
- MySQL / MariaDB (the capacity lock uses the `GET_LOCK()` function)

Installation steps

1. **Joomla Admin → System → Install → Extensions**
2. Upload `pkg_formexpiry_v1.9.0.zip`
3. The package installs three parts and enables all of them:
   - **System - Form Expiry for RSForm! Pro** (`plg_system_formexpiry`) — the field type itself
   - **Form Expiry - Status** (`mod_formexpiry_status`) — administrator dashboard, placed on the Home Dashboard automatically (section 15)
   - **Task - Form Expiry notifications** (`plg_task_formexpiry`) — scheduled notifications (section 16)
4. The field class is copied into the RSForm! Pro helpers folder automatically

The parts are managed together by the package: they cannot be uninstalled one by one, only the package (**System → Manage → Extensions**, type: Package).

Verifying the installation

```
-- Is the field type registered?
SELECT * FROM `#__rsform_component_types` WHERE ComponentTypeId = 703;

-- Are the properties defined?
SELECT COUNT(*) FROM `#__rsform_component_type_fields` WHERE ComponentTypeId = 703;
-- Expected result: 32
```

If the field does not appear in the RSForm! Pro Add Field list:

- **System → Manage → Plugins** → search for `Form Expiry` → enable it
- **System → Clear Cache**

---

3. File Structure

```
pkg_formexpiry_v1.9.0.zip
├── pkg_formexpiry.xml                    # Package manifest (v1.9.0, update server)
├── script.php                            # Package installer (module on the Home
Dashboard, removes the old update site)
├── language/{en-GB,hu-HU}/pkg_formexpiry.sys.ini
└── packages/
    ├── plg_system_formexpiry.zip
    ├── mod_formexpiry_status.zip
    └── plg_task_formexpiry.zip
```

plg_system_formexpiry (the field type):

```
plg_system_formexpiry/
├── formexpiry.xml                          # Plugin manifest
├── formexpiry.php                          # Main plugin class (hooks, static API, AJAX)
├── helper.php                              # FormexpiryHelper - shared logic (plugin,
module, task)
├── script.php                              # Install/update script
├── CHANGELOG.md                            # Changelog
├── admin/helpers/fields/formexpiry.php     # Field class (extends RSFormProField)
├── language/{en-GB,hu-HU}/                 # Language files
├── media/css/formexpiry.css, admin.css     # Styles
├── media/js/formexpiry.js                  # Frontend JavaScript
└── sql/mysql/                              # install.sql, uninstall.sql, update_*.sql (…
update_1.9.0.sql)
```

mod_formexpiry_status (administrator dashboard):

```
mod_formexpiry_status/
├── mod_formexpiry_status.xml
├── services/provider.php
├── src/Dispatcher/Dispatcher.php           # Permission check, data for the layout
├── src/Helper/StatusHelper.php             # Aggregation, filtering, sorting
├── tmpl/default.php                        # Layout (override:
administrator/templates/atum/html/mod_formexpiry_status/)
├── media/css/status.css
├── media/js/status.js                      # Close / Reopen buttons (1.9.0)
└── language/{en-GB,hu-HU}/
```

plg_task_formexpiry (scheduled notifications):

```
plg_task_formexpiry/
├── formexpiry.xml
├── services/provider.php
├── src/Extension/Formexpiry.php            # "formexpiry.notify" task type
└── language/{en-GB,hu-HU}/
```

---

4. Adding and Configuring the Field

Adding the field

1. **RSForm! Pro admin → select the form → Components tab**
2. Click **Add Field**
3. **Advanced** category → **Form Expiry** (⏰ icon)
4. Drag it into the form

Minimal configuration (in 5 minutes)

VALIDATION TAB:

```
Validation Message: Sorry, this form has expired and is no longer accepting submissions.
```

ATTRIBUTES TAB:

```
Expiry Date:               2026-12-31 23:59
Show Message in Field:     YES
Show Notification Popup:   YES
Days Before Expiry:        7
```

With this setup the form shows "This form will close in X days!" during the last 7 days, then closes at 2026-12-31 23:59 and displays the Validation Message.

Opening date (registration start)

When `OPEN_DATE` is set, the form is **"not open yet"** until that time:

- the server rejects submissions (also with JavaScript off and for tampered requests),
- the submit buttons are disabled and a blue message is shown in the field (`PENDING_MESSAGE`, default: "Registration opens on {OPEN_DATE}."),
- the popup and the striped overlay can be shown as well (same settings as for expiry),
- at the opening moment the form **opens in place**: buttons are enabled, the message and overlay disappear — no page reload needed.

The opening date is interpreted in the same timezone as the expiry date (see section 6). It also works without an expiry date: the form then stays open after opening (or until the capacity is reached).

The browser **synchronises the opening moment with the server clock**: on page load a small, uncached request (`com_ajax`) fetches the server time, so a computer clock that runs fast cannot enable the button early. If this request fails, the browser's own clock is used — the server still checks every submission against its own clock.

Example configurations

Registration that opens at a set time (limited places)

```
Open Date:                 2026-11-02 09:00
Expiry Date:               2026-11-20 23:59
Not-yet-open Message:      Registration opens on {OPEN_DATE}.
Enable Capacity Limit:     YES
Max Submissions:           40
Countdown Mode:            live
Show Form Overlay Stripes: YES
```

Before opening, the live countdown counts down to the opening ("Opens in: 2 days 4 hours …"); at opening it switches to the expiry ("Expires in: …").

Form reused every year

```
Open Date:                           2027-01-10 08:00
Expiry Date:                         2027-01-31 23:59
Enable Capacity Limit:               YES
Max Submissions:                     100
Count Submissions From Opening Date:  YES
```

Last year's submissions do not have to be deleted: only submissions received since the opening date count towards the capacity.

Event registration (7-day warning)

```
Expiry Date:               2026-03-15 23:59
Validation Message:        Registration for this event has closed.
Show Message in Field:     YES
Show Notification Popup:   YES
Days Before Expiry:        7
Hours Before Expiry:       0
```

Flash sale (hourly warning)

```
Expiry Date:               2026-02-06 18:00
Validation Message:        The promotion has ended!
Show Notification Popup:   NO
Days Before Expiry:        0
Hours Before Expiry:       6
```

Registration with limited capacity

```
Expiry Date:               2026-12-31 23:59
Enable Capacity Limit:     YES
Max Submissions:           50
Show Capacity Counter:     YES
Validation Message:        There are no places left for this event.
```

Course with 30 places, a waiting list of 10 and one submission per email (1.9.0)

```
Enable Capacity Limit:     YES
Max Submissions:           30
Waiting List Size:         10
Waiting List Message:      All places are taken – {WAITLIST_FREE} more people can join
the waiting list.
Unique Field:              Email
Duplicate Message:         You have already registered with this email address.
```

Put the `{formExpiry:status}` placeholder into the confirmation email (replace `formExpiry` with the name of your field): applicants 31–40 get "Waiting list", everyone else "Place confirmed" (section 8).

---

5. Properties Reference

From 1.9.0 the settings on the **Attributes** tab are grouped: dates and timezone → messages and warning → capacity and waiting list → one submission per person → display → advanced. Dependent settings are only shown when their switch is on:

Switch	When on, shows
`ENABLE_CAPACITY_LIMIT = YES`	`MAX_SUBMISSIONS`, `CAPACITY_COUNT_FROM_OPEN`, `CAPACITY_COUNT_CONFIRMED_ONLY`, `SHOW_CAPACITY_COUNTER`, `WAITLIST_SIZE`, `WAITLIST_MESSAGE`
`SHOW_NOTIFICATION_POPUP = YES`	`POPUP_AUTO_CLOSE_DELAY`, `POPUP_DEV_MODE`
`TIMEZONE_MODE = fixed`	`FIXED_TIMEZONE`

Hidden settings keep their values; they are just not shown. A summary of the settings is also shown in the field list (section 12).

GENERAL TAB

Property	Type	Description
`NAME`	textbox	Unique field identifier (e.g. `formExpiry`)
`CAPTION`	textbox	Caption (usually left empty — the field is hidden)

VALIDATION TAB

Property	Type	Description
`REQUIRED`	hidden	Always NO — not meaningful for this field type. From 1.9.0 it is no longer shown among the settings.

Property	Type	Description
`VALIDATIONMESSAGE`	textarea	Message shown when the form expires by date. If left empty, the default text is used ("Sorry, this form has expired…"). Multilingual — set it in every active language. Before opening, when the capacity is reached or when the form is closed dynamically, a different message is shown (see sections 4, 8 and 13).

ATTRIBUTES TAB — Basic settings

Property	Type	Default	Description
`OPEN_DATE`	textbox	*(empty)*	**1.7.0.** Opening date: the form cannot be submitted before this time. Same format as `EXPIRY_DATE`. Leave empty for no opening condition. If it is not earlier than the expiry date, the admin preview shows a warning.
`EXPIRY_DATE`	textbox	`2026-12-31 23:59`	Expiry date and time. Format: `YYYY-MM-DD HH:MM` (also accepted: `YYYY-MM-DD HH:MM:SS`, `YYYY-MM-DD`). If the date does not exist on the calendar (e.g. `2026-02-30`), the date condition is ignored and the admin preview shows "Invalid date".
`SUBMIT_BUTTON_SELECTORS`	textarea	—	One CSS selector per line for custom submit buttons. Built-in: `button[type="submit"]`, `input[type="submit"]`, `button.rsform-submit-button`
`SHOW_MESSAGE_IN_FIELD`	select (YES/NO)	YES	Show a message inside the field container (the not-yet-open message before opening, the "will close in X days" text during the warning period, the expiry message once expired)
`SHOW_NOTIFICATION_POPUP`	select (YES/NO)	YES	Show a popup before opening, in the expired state (and, during the warning period, on page load). At opening the popup closes by itself.
`POPUP_AUTO_CLOSE_DELAY`	textbox	`5`	Delay in seconds before the popup closes automatically (`0` = never closes automatically)

Property	Type	Default	Description
`POPUP_DEV_MODE`	select (NO/YES)	NO	Developer mode: the popup does not close automatically, and detailed logging appears in the browser console (e.g. the measured clock offset). It does not trigger any state — for testing, see section 13.
`SHOW_FORM_OVERLAY_STRIPES`	select (NO/YES)	NO	Striped overlay on a form that cannot be submitted (not yet open / expired / full)
`USE_SEPARATE_EXPIRED_MESSAGE`	select (NO/YES)	NO	When the form expires by date, show the `PLG_FORMEXPIRY_EXPIRED_MESSAGE` language string instead of the Validation Message (customizable with a language override)
`PENDING_MESSAGE`	textarea	*(empty)*	**1.7.0.** Message shown before the opening date. `{OPEN_DATE}` is replaced with the formatted opening date (using `STATIC_DATE_FORMAT`; in `user_browser` / `fixed` mode in the visitor's local time). If left empty: `PLG_FORMEXPIRY_DEFAULT_PENDING_MESSAGE`.
`DAYS_BEFORE_EXPIRY`	select	`0`	How many days before expiry the warning appears. Values: 0, 1, 2, 3, 5, 7, 14, 30
`HOURS_BEFORE_EXPIRY`	select	`0`	How many hours before expiry the warning appears. Values: 0, 1, 2, 6, 12, 24, 48, 72. Added to the days!

ATTRIBUTES TAB — Capacity limit

Property	Type	Default	Description
`ENABLE_CAPACITY_LIMIT`	select (NO/YES)	NO	Enable the capacity limit
`MAX_SUBMISSIONS`	textbox	`100`	Maximum number of accepted submissions. Only a positive whole number is valid; any other value is reported as a configuration error (1.9.0), see section 13.

Property	Type	Default	Description
`CAPACITY_COUNT_FROM_OPEN`	select (NO/YES)	NO	**1.7.0.** If YES and `OPEN_DATE` is set, only submissions received since the opening date count towards the capacity
`SHOW_CAPACITY_COUNTER`	select (YES/NO)	YES	Show the "X / Y spots remaining" counter
`CAPACITY_COUNT_CONFIRMED_ONLY`	select (NO/YES)	NO	**1.9.0.** If YES, only confirmed submissions count towards the capacity (RSForm! Pro submission confirmation must be enabled for the form)

ATTRIBUTES TAB — Waiting list and one submission per person (1.9.0)

Property	Type	Default	Description
`WAITLIST_SIZE`	textbox	`0`	After the maximum is reached, this many further submissions are accepted onto the waiting list (`0` = no waiting list). Only works together with the capacity limit.
`WAITLIST_MESSAGE`	textarea	*(empty)*	Message shown on the form when all places are taken but the waiting list is still open. Placeholders: `{WAITLIST_FREE}`, `{WAITLIST_SIZE}`. If left empty: `PLG_FORMEXPIRY_DEFAULT_WAITLIST_MESSAGE`.
`UNIQUE_FIELD`	textbox	*(empty)*	The name of a form field (e.g. `Email`). Only one submission is accepted per value (ignoring case and surrounding spaces). Can be used without a capacity limit.
`ONE_PER_USER`	select (NO/YES)	NO	A logged-in user can submit the form only once. Does not apply to guests.
`DUPLICATE_MESSAGE`	textarea	*(empty)*	Message shown on a repeated submission. If left empty: `PLG_FORMEXPIRY_DEFAULT_DUPLICATE_MESSAGE`.

ATTRIBUTES TAB — Countdown

Property	Type	Default	Description
`COUNTDOWN_MODE`	select	`off`	`off` = no countdown / `static` = formatted date text / `live` = live countdown with seconds
`STATIC_DATE_FORMAT`	select	`short`	Date format for the static display and for the `{OPEN_DATE}` placeholder: `short` / `medium` / `long` / `human`

Static date format examples (English):

- `short` → `2026-12-31 23:59`
- `medium` → `December 31, 2026 - 23:59`
- `long` → `Thursday, December 31, 2026 - 23:59`
- `human` → `December 31, 2026`

The formats come from the language file (`PLG_FORMEXPIRY_DATE_FORMAT_*`) and can be changed with a language override.

ATTRIBUTES TAB — Timezone

Property	Type	Default	Description
`TIMEZONE_MODE`	select	`server`	`server` / `user_browser` / `fixed`
`FIXED_TIMEZONE`	select	`UTC`	IANA timezone name (e.g. `Europe/London`) — only with `TIMEZONE_MODE=fixed` . If the value is invalid, the Joomla timezone is used and the admin preview shows a warning.

Other

Property	Type	Description
`DESCRIPTION`	textarea	Admin description
`COMPONENTTYPE`	hidden	`703` — cannot be changed

---

6. Timezone Modes

> Timezone mode only matters when a date is set in `OPEN_DATE` or `EXPIRY_DATE`. Both dates are interpreted in the same timezone.

The key principle: one deadline, one moment

In every mode a deadline is **a single moment in time**: the form opens and closes for all visitors at the same instant, wherever they are. The three modes only decide two things:

- **in which timezone you enter** the dates, and
- **in which timezone visitors see** them (in the static date display and in `{OPEN_DATE}`).

It is not possible to close one form "at local midnight" in each visitor's own timezone. If you need separate regional deadlines, use a separate form per region (for example with Fixed Timezone).

Comparison

Aspect	Server Time	User Browser Time	Fixed Timezone
Admin enters the date in	Site timezone	Site timezone	The timezone you choose
Visitors see	The same time for everyone	Their own local time	Their own local time
Reference timezone	Site	Site	Chosen by the admin
JavaScript needed for display	No	Yes	Yes
Typical use	Local, simple	Global, automatic	Global, tied to a venue or market
DST handling	Automatic	Automatic	Automatic

"Site timezone" is the Website Time Zone in Joomla Global Configuration (PHP's default timezone does not matter). The examples below assume the site timezone is `Europe/Budapest`.

1. Server Time (default)

The dates are interpreted in the site timezone and shown exactly as entered, to everyone.

```
EXPIRY_DATE:    2026-12-31 20:00
Site timezone:  Europe/Budapest

Visitor in Budapest sees:  2026-12-31 20:00
Visitor in New York sees:  2026-12-31 20:00   (it is 14:00 there when the form closes)
Visitor in Tokyo sees:     2026-12-31 20:00   (it is 04:00 on Jan 1 there)
```

When to use: local events, a mostly local audience, or when you want the simplest setup. **Keep in mind:** visitors in other timezones have to convert the time themselves — mention the timezone in the form text if your audience is international.

2. User Browser Time

The dates are interpreted in the site timezone, but the browser converts the displayed date to **each visitor's local time**.

```
EXPIRY_DATE:    2026-12-31 20:00
Site timezone:  Europe/Budapest

Visitor in Budapest sees:  2026-12-31 20:00
Visitor in New York sees:  2026-12-31 14:00
Visitor in Tokyo sees:     2027-01-01 04:00
```

The browser's own timezone setting is used (`Intl.DateTimeFormat().resolvedOptions().timeZone`). No cookie or session is involved: the conversion happens instantly on the first visit and also works on cached pages. Without JavaScript the date is shown in the site timezone.

When to use: online events and webinars with an international audience, when the site timezone is also the natural reference.

3. Fixed Timezone

You choose the reference timezone in `FIXED_TIMEZONE` ; the dates are interpreted in that timezone, and visitors see them converted to their own local time.

```
EXPIRY_DATE:     2026-12-31 20:00
FIXED_TIMEZONE:  Europe/Budapest
Site timezone:   America/New_York   (e.g. US hosting)

Visitor in Budapest sees:  2026-12-31 20:00
Visitor in New York sees:  2026-12-31 14:00
Visitor in Tokyo sees:     2027-01-01 04:00
```

Difference from Server Time: in Server Time mode the date is always entered in the site timezone and nobody gets a conversion. With Fixed Timezone you enter the date in the event's own timezone, regardless of where the server is, and every visitor gets their local time. Without JavaScript the date is shown in the fixed timezone.

When to use: an event at a physical venue with an online stream, a campaign tied to one market ("EU launch at 9:00 CET"), or when the server's timezone differs from the event's.

The list offers more than 120 IANA timezones, for example `Europe/London`, `Europe/Budapest`, `America/New_York`, `America/Los_Angeles`, `Asia/Tokyo`, `Asia/Dubai`, `Australia/Sydney`, `Pacific/Auckland`. Non-existent zones that appeared in the list of earlier versions (`Europe/Munich`, `Europe/Milan`, `Europe/Barcelona`, `Europe/Ankara`, `America/Rio_de_Janeiro`) and outdated names (`Europe/Kiev`, `Asia/Calcutta`, etc.) are mapped automatically to the correct valid zone, so forms that were already configured keep working.

Which mode to choose

```
Where does the event take place?
│
├── At a physical location (one city or country)
│     ├── No online participation  → SERVER TIME
│     └── Also online / streamed   → FIXED TIMEZONE (the venue's timezone)
│
└── Online only / global audience  → USER BROWSER TIME
      (or FIXED TIMEZONE if the reference time is not the site's timezone)
```

Common scenarios

Scenario	Mode	Why
Local shop, offer valid until 31 December	Server Time	Customers are local; simplest
Webshop with a worldwide sale	User Browser Time	Everyone sees the closing time in their own local time. Note: the sale ends at the same moment everywhere, not at each visitor's local midnight.
Product launch campaign in the EU	Fixed Timezone (`Europe/Brussels`)	The deadline is tied to a market's time, but international visitors see their local time
Online course, registration opens at 10:00 Budapest time	Fixed Timezone (`Europe/Budapest`)	The instructor's time is the reference; students abroad see their local time

Daylight saving time

Opening and expiry are calculated as Unix timestamps from the date and the timezone, so DST transitions are handled automatically in every mode: a deadline of 20:00 in summer and one of 20:00 in winter both close at 20:00 local time in the reference timezone.

---

7. Countdown Timer

Modes

`COUNTDOWN_MODE = off` — No countdown is shown (default).

`COUNTDOWN_MODE = static` — The date is shown as formatted text. Before opening it shows the opening date ("Opens in: November 2, 2026 - 09:00"), afterwards the expiry date ("Expires in: …"). The format is set with the `STATIC_DATE_FORMAT` property.

`COUNTDOWN_MODE = live` — Live countdown, updated every second. Before opening it counts down to the opening (prefix: `PLG_FORMEXPIRY_OPEN_PREFIX`), at opening it switches to the expiry (prefix: `PLG_FORMEXPIRY_COUNTDOWN_PREFIX`); if there is no expiry date, it disappears at opening. Every state change happens **in place** (no page reload, anything the visitor has typed is kept). This also works without a countdown.

Live countdown HTML structure

```
<div class="formexpiry-live-countdown">             <!-- before opening: + formexpiry-
live-countdown--pending -->
  <span class="countdown-prefix">Expires in: </span>  <!-- before opening: "Opens in: " -
->
  <span class="countdown-time">
    <span class="countdown-unit countdown-days">
      <span class="countdown-value">5</span>
      <span class="countdown-label">days</span>
    </span>
    <span class="countdown-unit countdown-hours">...</span>
    <span class="countdown-unit countdown-minutes">...</span>
    <span class="countdown-unit countdown-seconds">...</span>
  </span>
  <span class="countdown-suffix"></span>
</div>
```

---

8. Capacity Limit

With `ENABLE_CAPACITY_LIMIT = YES`, the plugin checks the current number of submissions in real time from the `#__rsform_submissions` table, filtered by the form's `FormId`.

With `CAPACITY_COUNT_FROM_OPEN = YES` and an `OPEN_DATE`, only submissions received since the opening date are counted (`DateSubmitted >= opening`). This way a form reopened every year does not need its old submissions deleted. In this mode every opening period counts as a separate notification event (section 9).

With `SHOW_CAPACITY_COUNTER = YES`, the frontend shows a counter:

- While places are available: "X / Y spots remaining" (X = number of free places)

- When full, the counter disappears and the expiry message is shown: "The form has reached its maximum capacity (Y submissions) and has been closed."

The texts can be changed in the language file (`PLG_FORMEXPIRY_CAPACITY_COUNTER`, `PLG_FORMEXPIRY_CAPACITY_LIMIT_REACHED`).

Capacity-based closure is also enforced by **server-side validation** (the field's `processValidation()` method, with a fresh count at the moment of submission), not only in JavaScript.

Simultaneous submissions (1.7.0): on forms with a capacity limit, counting and saving are serialised (MySQL `GET_LOCK`, per form). Even if many people submit at the exact moment of opening, the limit is not exceeded. The lock is released right after the submission is saved — and immediately when validation fails (1.9.0). If it cannot be acquired within 10 seconds, the server **rejects** the submission with a "please try again" message (1.9.0; earlier versions let it through without the lock, so the limit could be exceeded), and an entry is written to the Joomla log (category `formexpiry`).

Confirmed submissions only (1.9.0)

By default every submission counts, unconfirmed ones included — so anyone can fill up a public registration with junk submissions. With `CAPACITY_COUNT_CONFIRMED_ONLY = YES` only those submissions hold a place that the applicant has confirmed through the RSForm! Pro confirmation email (submission confirmation — Confirm Submission — must be enabled in the form's properties).

Keep in mind: applicants who have not confirmed yet do not hold a place. If several people apply at the same time and all confirm later, the number of participants can slightly exceed the limit — handle this with a waiting list or a limit set a few places lower.

One submission per person (1.9.0)

- **`UNIQUE_FIELD`** — enter the name of a form field (typically the email field). If a submission with the same value already exists, the server rejects the new one and shows the `DUPLICATE_MESSAGE` text. The comparison ignores letter case and surrounding spaces.
- **`ONE_PER_USER`** — one submission per logged-in user. It has no effect on guests; `UNIQUE_FIELD` covers them.

Both can be used together. With `CAPACITY_COUNT_FROM_OPEN = YES`, uniqueness also applies only to submissions received since the opening date (on a form reused every year, last year's applicants can apply again). The check runs serialised with the capacity lock, so of two simultaneous submissions with the same email address only one gets through. On forms with AJAX validation the check happens on the actual submission.

This protection filters out accidental double submissions and simple abuse; it does not stop someone who deliberately uses a different email address each time (that needs confirmation or login).

Waiting list (1.9.0)

With `WAITLIST_SIZE = 10` and `MAX_SUBMISSIONS = 30` the form stays open up to 40 submissions. After the 30th submission:

- the waiting list message (`WAITLIST_MESSAGE`) appears on the form, and the counter shows the free waiting list places;
- the admin notification email is sent at this point (when the main capacity is reached); no further email is sent when the waiting list fills up;
- after the 40th submission the form closes ("full").

Who is on the waiting list? The order is given by the time of submission: the first 30 submissions have a place, the rest are on the waiting list. Applicants learn this from the emails and the Thank You message through the following placeholders (`fieldname` is the name of the Form Expiry field):

Placeholder	Value
`{fieldname:status}`	"Place confirmed" or "Waiting list" (`PLG_FORMEXPIRY_PLACE_CONFIRMED` / `PLG_FORMEXPIRY_PLACE_WAITLIST` , changeable with a language override)
`{fieldname:position}`	The position of the submission (e.g. `34`)
`{fieldname:waitlist_position}`	Position on the waiting list (e.g. `4`); empty for a confirmed place

The position is **dynamic**: if you delete an earlier submission (a cancellation), everyone behind it moves up by one, and number 31 automatically gets a place. The plugin does not send a separate notification about this — in the submissions list the order by submission date shows who is in; you have to notify the person yourself (e.g. with RSForm! Pro's "Resend Emails" function, which already writes the new status into the email).

---

9. Admin Email Notifications

Email notifications are configured in the plugin settings (not in the field properties).

Admin → System → Plugins → Form Expiry → plugin parameters:

Setting	Description
`enable_email_notifications`	Turn email notifications on (YES/NO)

Setting	Description
`admin_email_addresses`	Recipient addresses (one per line; commas and semicolons are also accepted)
`email_subject`	Email subject. Placeholder: `{FORM_NAME}`
`email_body`	Email body. Placeholders: `{FORM_NAME}`, `{FORM_ID}`, `{REASON}`, `{EXPIRY_DATE}`, `{CURRENT_SUBMISSIONS}` (alias: `{SUBMISSION_COUNT}`), `{MAX_SUBMISSIONS}`. If left empty, the default text from the language file is sent.
`notification_type`	When to notify: `on_expiry` / `on_capacity` / `both`
`invalid_config_behavior`	**1.9.0.** On invalid configuration (invalid date or capacity): `The form stays open` (default) / `The form is closed` — see section 13

Emails are sent **once per event**, regardless of how many visitors open the form (database-based deduplication in the `#__formexpiry_notifications` table). If you change the expiry date or the capacity — or, with `CAPACITY_COUNT_FROM_OPEN`, the opening date — that counts as a new event and a new notification can be sent. Dynamic closure and opening do not send an email. With a waiting list (1.9.0) the capacity notification is sent when the main capacity is reached. If the deduplication table cannot be reached (database error), the plugin does not send an email but writes a log entry (1.9.0) — so even in a broken state it cannot flood your inbox.

The notification is triggered the first time the expired form is opened or a submission is attempted on it — or (from 1.8.0) the next time the scheduled task runs (section 16). If nobody opens the form, the email is only sent by the scheduled task.

---

10. CSS Customization

CSS class reference

Class	Description	Default style
`.formexpiry-field-container`	Main field container	—

Class	Description	Default style
`.formexpiry-pending-message`	"Not open yet" message (1.7.0)	Blue background, 🕑 icon
`.formexpiry-warning-message`	Warning message	Yellow background
`.formexpiry-expired-message`	Expiry message	Red background
`.formexpiry-rejected-message`	Reason of a rejected submission (e.g. "already registered"), used together with `.formexpiry-expired-message` (1.9.0)	Red background
`.formexpiry-waitlist-message`	Waiting list message (1.9.0)	Yellow background, 📋 icon
`.formexpiry-notification`	Popup container	White, centred, shadow
`.formexpiry-notification-overlay`	Dimmed background behind the popup	Semi-transparent dark
`.formexpiry-notification-close`	Popup close button	Top right, ×
`.formexpiry-notification-dev`	Developer mode notice inside the popup	Yellow box
`.formexpiry-form-overlay`	Striped overlay on the form (not yet open / expired / full)	Diagonal stripes
`.formexpiry-capacity-counter`	Capacity counter	—
`.formexpiry-static-countdown`	Static date container	Light grey background, border
`.formexpiry-live-countdown`	Live countdown container	Light grey background, border

Class	Description	Default style
`.formexpiry-live-countdown--pending`	Live countdown before opening (1.7.0)	— (for your own styling)
`.countdown-prefix`	Countdown prefix text	Semi-bold
`.countdown-time`	Container of the countdown values	Bold, dark text
`.countdown-suffix`	Countdown suffix text	Normal
`.countdown-unit`	One time unit (days/hours/minutes/seconds)	Inline-block, spacing
`.countdown-value`	Numeric value	Extra bold, 16px
`.countdown-label`	Unit name	Normal, grey, 13px
`.countdown-days` / `.countdown-hours` / `.countdown-minutes` / `.countdown-seconds`	Individual units	—

The HTML structure of the live countdown is shown in section 7. The live countdown shows seconds only when less than a day is left, and hides units that are zero (e.g. "2 hours 5 seconds").

Where to put your CSS

1. **For one form only — RSForm! Pro form properties.** In RSForm! Pro open the form → **Properties** → **CSS and JavaScript**, and add your rules there. This is the easiest option when you only want to style one form.
2. **For the whole site — your template's custom CSS file.** With Cassiopeia (Joomla 4.1+) create `media/templates/site/cassiopeia/css/user.css`; it is loaded automatically. Other templates usually offer a custom CSS file or a "Custom CSS" field in the template style options.
3. **Replacing the plugin's stylesheet completely — template override.** Copy `media/plg_system_formexpiry/css/formexpiry.css` to:
   - Joomla 4.1+ templates (e.g. Cassiopeia): `media/templates/site/YOUR_TEMPLATE/css/plg_system_formexpiry/formexpiry.css`
   - Older-style templates: `templates/YOUR_TEMPLATE/css/plg_system_formexpiry/formexpiry.css`

Joomla then loads your copy **instead of** the plugin's file, and plugin updates no longer overwrite it. Because it replaces the whole stylesheet (popup, overlay, messages and countdown), start from a copy of the original, and compare it with the new original after major plugin updates. **When updating to 1.7.0**, an existing override will be missing the `.formexpiry-pending-message` style, and **when updating to 1.9.0** the `.formexpiry-waitlist-message` style — copy them from the new original.

Specificity tip: the plugin's stylesheet may load after your template CSS. The examples below therefore start with `.formexpiry-field-container`, which makes them slightly more specific than the plugin's defaults, so they win without `!important`.

Customization examples

Example 1: Coloured numbers per unit

```
.formexpiry-field-container .formexpiry-live-countdown .countdown-days .countdown-value
{ color: #dc3545; }
.formexpiry-field-container .formexpiry-live-countdown .countdown-hours .countdown-value
{ color: #fd7e14; }
.formexpiry-field-container .formexpiry-live-countdown .countdown-minutes .countdown-
value { color: #ffc107; }
.formexpiry-field-container .formexpiry-live-countdown .countdown-seconds .countdown-
value { color: #28a745; }
```

Example 2: Large digital style

```
.formexpiry-field-container .formexpiry-live-countdown {
    background: #000;
    color: #0f0;
    border: 2px solid #0f0;
    font-family: 'Courier New', monospace;
    padding: 15px;
}

.formexpiry-field-container .formexpiry-live-countdown .countdown-time,
.formexpiry-field-container .formexpiry-live-countdown .countdown-value {
    font-size: 32px;
    font-weight: 700;
    color: #0f0;
}

.formexpiry-field-container .formexpiry-live-countdown .countdown-label {
    font-size: 14px;
    color: #0f0;
    opacity: 0.7;
}
```

Example 3: Compact badge

```
.formexpiry-field-container .formexpiry-live-countdown {
    display: inline-block;
    background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);
    color: #fff;
    border: none;
    border-radius: 20px;
    padding: 6px 12px;
    font-size: 13px;
}

.formexpiry-field-container .formexpiry-live-countdown .countdown-unit {
    margin-right: 4px;
}

.formexpiry-field-container .formexpiry-live-countdown .countdown-time,
.formexpiry-field-container .formexpiry-live-countdown .countdown-value {
    font-size: 14px;
    color: #fff;
}

.formexpiry-field-container .formexpiry-live-countdown .countdown-label {
    font-size: 11px;
    color: rgba(255, 255, 255, 0.8);
}
```

Example 4: Vertical stack

```
.formexpiry-field-container .formexpiry-live-countdown {
    text-align: center;
}

.formexpiry-field-container .formexpiry-live-countdown .countdown-unit {
    display: block;
    margin: 0 0 8px;
}

.formexpiry-field-container .formexpiry-live-countdown .countdown-value {
    display: block;
    font-size: 24px;
    margin-bottom: 2px;
}

.formexpiry-field-container .formexpiry-live-countdown .countdown-label {
    display: block;
    font-size: 11px;
    text-transform: uppercase;
}
```

Example 5: Box grid

```
.formexpiry-field-container .formexpiry-live-countdown .countdown-unit {
    display: inline-block;
    background: #fff;
    border: 1px solid #dee2e6;
    border-radius: 4px;
    padding: 8px 12px;
    margin-right: 8px;
    text-align: center;
}

.formexpiry-field-container .formexpiry-live-countdown .countdown-value {
    display: block;
    font-size: 20px;
    line-height: 1;
    margin-bottom: 4px;
}

.formexpiry-field-container .formexpiry-live-countdown .countdown-label {
    display: block;
    font-size: 10px;
    text-transform: uppercase;
    letter-spacing: 0.5px;
}
```

Example 6: Highlighting the countdown before opening (1.7.0)

```
.formexpiry-field-container .formexpiry-live-countdown--pending {
    background: #cfe2ff;
    border-color: #0d6efd;
}

.formexpiry-field-container .formexpiry-live-countdown--pending .countdown-value {
    color: #084298;
}
```

Messages and popup

```
/* Not-yet-open message */
.formexpiry-field-container .formexpiry-pending-message {
    background: #e7f1ff;
    border-color: #0d6efd;
}

/* Warning message */
.formexpiry-field-container .formexpiry-warning-message {
    background: #fff3cd;
    border-color: #ffc107;
    border-radius: 8px;
}

/* Expiry message */
.formexpiry-field-container .formexpiry-expired-message {
    background: #f8d7da;
    border-color: #dc3545;
}

/* Popup (it is placed directly in <body>, outside the field container) */
body .formexpiry-notification {
    border-radius: 12px;
    box-shadow: 0 10px 40px rgba(0, 0, 0, 0.3);
}
```

Changing the texts

All visible texts come from the language files and can be changed without editing plugin files, using **System → Manage → Language Overrides** (Site). The most common keys:

Key	Default (English)
`PLG_FORMEXPIRY_DEFAULT_PENDING_MESSAGE`	`Registration opens on {OPEN_DATE}.`
`PLG_FORMEXPIRY_OPEN_PREFIX`	`Opens in:`
`PLG_FORMEXPIRY_COUNTDOWN_PREFIX`	`Expires in:`
`PLG_FORMEXPIRY_COUNTDOWN_SUFFIX`	*(empty)*
`PLG_FORMEXPIRY_COUNTDOWN_DAYS` / `_HOURS` / `_MINUTES` / `_SECONDS`	`days` / `hours` / `minutes` / `seconds`
`PLG_FORMEXPIRY_WARNING_DAYS` / `_HOURS` / `_SOON`	`This form will close in %s days!` …
`PLG_FORMEXPIRY_CAPACITY_COUNTER`	`%d / %d spots remaining`

Key	Default (English)
`PLG_FORMEXPIRY_DATE_FORMAT_SHORT` / `_MEDIUM` / `_LONG` / `_HUMAN`	Date formats (PHP date syntax)

For shorter unit labels, override the `PLG_FORMEXPIRY_COUNTDOWN_*` keys (e.g. `d`, `h`, `m`, `s`).

Tips

1. Use the browser's developer tools (F12) to inspect the countdown's HTML and try out changes live.
2. Keep your selectors specific (start them with `.formexpiry-field-container`) to avoid conflicts with the template.
3. Use `!important` only if a rule still does not apply.

---

11. Frontend Output

Not open yet (1.7.0)

```
<div class="formexpiry-field-container">
  <div class="formexpiry-pending-message" role="status">Registration opens on 2026-11-02
09:00.</div>
  <input type="hidden" data-status="pending" data-open-timestamp="1793606400" ... />
</div>
```

Active state

```
<div class="formexpiry-field-container">
  <input type="hidden" name="form[formExpiry]"
    data-status="active"
    data-expiry-timestamp="1798757940"
    value="2026-12-31 23:59" />
</div>
```

Warning state

The text is generated automatically (language strings `PLG_FORMEXPIRY_WARNING_DAYS` / `_HOURS` / `_SOON`) and is refreshed every minute while the page is open.

```
<div class="formexpiry-field-container">
  <div class="formexpiry-warning-message" role="status">This form will close in 2 days!
</div>
  <input type="hidden" data-status="warning" ... />
</div>
```

Expired state

```
<div class="formexpiry-field-container">
  <div class="formexpiry-expired-message" role="alert">Sorry, this form has expired...
</div>
  <input type="hidden" data-status="expired" ... />
</div>
```

Submit buttons before opening and in the expired state

- `disabled` attribute
- `opacity: 0.6`
- `cursor: not-allowed`
- Clicks and the form's `submit` event (Enter key, programmatic submission) are blocked in JavaScript
- The server also rejects the submission (`processValidation()`)
- At opening the plugin restores the buttons — unless another Form Expiry field on the same form is still blocking

JavaScript hook for custom logic

When the state changes (e.g. the form opens or expires while the page is open), the field dispatches a `formexpiry:statuschange` event (it bubbles). At opening, `previous` is `"pending"`:

```
document.addEventListener('formexpiry:statuschange', function(e) {
    if (e.detail.previous === 'pending') {
        // e.g. focus the first field when the form opens
    }
    console.log(e.detail.status, e.detail.previous, e.detail.field);
});
```

Fields that are already expired on page load:

```
document.addEventListener('DOMContentLoaded', function() {
    var expiredFields = document.querySelectorAll('input[data-status="expired"]');
    expiredFields.forEach(function(field) {
        // Your own logic...
    });
});
```

---

12. Admin Preview

In the RSForm! Pro form editor's field list, the Form Expiry field's card shows a summary, so most settings can be checked without opening the field. From 1.9.0:

Line 1 – state and next change, with the left bar and the icon in the state's colour:

State	Colour	Additional info
Not open yet	Blue	"opens in 5 h 20 min"
Active	Green	"closes in 3 d 4 h"
Expiring soon	Yellow	"closes in 1 d 2 h"
Waiting list	Yellow	the main capacity is full, the waiting list is still open
Expired	Red	"closed 2 d 1 h ago"
Full	Red	—
Closed	Red	closed manually or because of a configuration error
Invalid date	Red	—
Not configured	Grey	a short hint on what to set

Line 2 – dates: opening, expiry and the timezone they are interpreted in.

Line 3 – tags for the important settings that are switched on and for current numbers (only what is on): "Submissions: 27 / 30", "Waiting list: 4 / 10", "confirmed only", "counted from opening", "unique: Email", "1 per user", "warning 7 d before", "live countdown" / "static date", "popup", "striped overlay", "visitor's local time", "popup dev mode ON". Blue tags are current numbers or settings that deserve attention.

⚠ **Warnings** (in red), when something needs fixing:

- invalid opening / expiry date,
- the opening date is not earlier than the expiry date (the form could never be submitted),
- `MAX_SUBMISSIONS` is not a number,
- invalid fixed timezone,
- the field set as unique field does not exist on the form (e.g. a mistyped name) — the uniqueness check would not apply,
- the form is closed because of the configuration error (section 13).

Last line (grey): the reason of a manual closure and the number of blocked submission attempts in the last 30 days.

The preview is refreshed when the editor loads and when the field is saved; the numbers and the "… in" times reflect that moment.

---

13. Troubleshooting

The field does not appear in the RSForm! Pro Add Field list

1. Is the plugin enabled? **System** → **Manage** → **Plugins** → **Form Expiry**
2. Clear the cache: **System** → **Clear Cache**
3. Check the database: `SELECT * FROM #__rsform_component_types WHERE ComponentTypeId = 703;`

The submit buttons are not disabled

1. Browser console (F12) — any JavaScript errors?
2. Is `/media/plg_system_formexpiry/js/formexpiry.js` loaded?
3. For a custom submit button, add its selector to `SUBMIT_BUTTON_SELECTORS`

The message does not appear

- `SHOW_MESSAGE_IN_FIELD` = YES? (message inside the field)
- `SHOW_NOTIFICATION_POPUP` = YES? (popup)
- For the warning: is `DAYS_BEFORE_EXPIRY` or `HOURS_BEFORE_EXPIRY` set?

How do I test the states?

On a test form:

- **Opening:** set `OPEN_DATE` 2–3 minutes ahead, open the page and wait: at the opening moment the buttons are enabled without a reload.
- **Expired state:** set `EXPIRY_DATE` to a time in the past, or a few minutes ahead and wait with the page open (the form closes in place).
- **Warning:** set the expiry e.g. 2 days ahead and `DAYS_BEFORE_EXPIRY` to 3.
- **Server-side protection:** switch off JavaScript in the browser and try to submit the not-yet-open or expired form — it must be rejected.
- **Clock correction:** set your computer's clock a few minutes ahead; the form still opens according to the server time. With `POPUP_DEV_MODE = YES` the console shows the measured offset (`FormExpiry: clock offset …`).
- The colour of the admin preview shows the current state; `POPUP_DEV_MODE = YES` keeps the popup open.

Submission attempt before opening: which error message is shown?

Without JavaScript (or for a tampered request) the server rejects the submission. The field's own container then shows the not-yet-open (blue) message. From 1.9.0 the field's `{fieldname:validation}` placeholder also shows the actual reason (not open yet / full / closed / already registered) instead of the

date-expiry `VALIDATIONMESSAGE` text. The exception is AJAX validation: there RSForm! Pro takes the error message directly from the field settings, so on AJAX forms that use an opening date it is still worth giving `VALIDATIONMESSAGE` neutral wording (e.g. "This form is not accepting submissions right now.").

"Invalid date" in the admin preview

The `OPEN_DATE` or `EXPIRY_DATE` format is wrong, or the date does not exist (e.g. `2026-02-30`, `31.12.2026`). Correct format: `YYYY-MM-DD HH:MM`.

What happens in this case is decided by the plugin's **On invalid configuration** option (System → Plugins → Form Expiry, 1.9.0):

- **The form stays open** (default, the previous behaviour): the invalid condition is ignored — with a mistyped expiry date the form stays open indefinitely.
- **The form is closed**: the form does not accept submissions until you fix the error (visitors see a neutral "This form is not accepting submissions at the moment." message). This is the safer choice when keeping the deadline matters.

The same applies to a `MAX_SUBMISSIONS` value that is not a number. The dashboard lists forms with a configuration error first, marked "Configuration error".

Expiry is off by 1 hour (DST)

Handled automatically (opening and expiry are calculated in the Joomla configuration timezone or the fixed timezone, independently of PHP's default timezone). If you still see a difference, check:

- Joomla Global Configuration → Website Time Zone
- The `TIMEZONE_MODE` property
- With `FIXED_TIMEZONE`: does the admin preview show an "Invalid fixed timezone" warning?

User Browser Time: the local time is not shown

- The local-time conversion applies to the static date (`COUNTDOWN_MODE = static`) and to the `{OPEN_DATE}` placeholder; the live countdown does not depend on the timezone.
- Is JavaScript disabled in the browser?
- Check: browser console → `Intl.DateTimeFormat().resolvedOptions().timeZone`
- Without JavaScript the date is shown in the Joomla configuration timezone

Dynamic closure (from the dashboard or from code)

The easiest way is the dashboard's **Close** / **Reopen** button (section 15, 1.9.0). From code (e.g. an RSForm! Pro PHP script) use the API below.

By default the closure is **global**: it closes the form for every visitor (stored in the database) until you reopen it. Server-side validation takes it into account too, and it takes precedence over the opening date.

```
use Joomla\CMS\Plugin\PluginHelper;
PluginHelper::importPlugin('system', 'formexpiry');

// Close for everyone (empty message = default text)
plgSystemFormexpiry::closeForm($formId, 'Closed by the administrator');

// Close only for the current visitor (behaviour before 1.6.8)
plgSystemFormexpiry::closeForm($formId, 'You have already registered.', true);

// Check
$closed = plgSystemFormexpiry::isFormClosed($formId);
$reason = plgSystemFormexpiry::getCloseReason($formId);

// Reopen (clears both the global and the session closure)
plgSystemFormexpiry::reopenForm($formId);
```

> **Note:** the closure for the current visitor only (session) is a convenience, not a protection: the visitor removes it by clearing cookies. For a "one submission per person" rule use `UNIQUE_FIELD` / `ONE_PER_USER` (section 8).

---

14. Updating

Automatic updates

The update server is configured in the package manifest:

```
https://extensions.blaho.hu/updates/pkg_formexpiry.xml
```

Manual update steps

1. Back up your database
2. Upload the new `.zip` file through the Joomla installer
3. Clear the cache: **System → Clear Cache**

Updating to 1.7.0 does not change how existing fields behave: the new properties stay empty / `NO` until you set them.

Updating to 1.9.0

Install `pkg_formexpiry_v1.9.0.zip` over the existing version. The settings of existing fields are kept, and the new properties default to off (no waiting list, no uniqueness rule, every submission counts). What changes on its own:

- If the capacity lock cannot be acquired within 10 seconds, the server rejects the submission ("please try again") instead of letting it through without the lock.
- For a rejected submission the validation message shows the actual reason.
- The field value stored with the submission is always the server-side expiry date.
- The dashboard shows the Close / Reopen buttons (can be switched off) and the number of blocked attempts.
- The field settings are grouped, dependent settings are only shown when their switch is on, and the field list shows a richer summary (sections 5 and 12).
- A new table is created: `#__formexpiry_blocked`.

Consider the plugin's new **On invalid configuration** option (section 13).

Updating from 1.7.0 (or earlier) to the package

From 1.8.0 the extension is delivered as a **package**. Simply install the package zip (`pkg_formexpiry_v1.9.0.zip`) over the existing plugin — **do not uninstall the plugin first** (uninstalling would remove the Form Expiry fields, with their settings, from every form, and drop the closure and notification tables). The package takes over the existing plugin; settings and forms stay unchanged.

- The update server now belongs to the package: `https://extensions.blaho.hu/updates/pkg_formexpiry.xml`. The installer removes the plugin's old update site, so no duplicate update notifications appear.
- For scheduled notifications, create the task (section 16).

Version history (short)

Version	Date	Changes
1.9.0	2026-11-11	Security fixes (capacity lock, notifications, fail-open paths), capacity integrity (confirmed submissions, one submission per person), waiting list, Close / Reopen from the dashboard, blocked attempts log, summary in the admin field list, shorter settings list (details: CHANGELOG.md)
1.8.0	2026-10-15	Package (`pkg_formexpiry`): status dashboard administrator module, scheduled admin notifications (Joomla Scheduled Tasks), package-level update server
1.7.0	2026-10-15	Opening date (registration start), opening in place, countdown to opening, counting from the opening date, server clock sync, capacity lock against simultaneous submissions (details: CHANGELOG.md)

Version	Date	Changes
1.6.8	2026-10-13	Server-side blocking fixed, one email per event, automatic warning text, timezone list fixed, expiry without page reload, cache-safe JS, global dynamic closure, accessible popup
1.6.7	2026-07-04	Performance, debug messages removed, idempotent property sync
1.6.6	2026-05-24	Timezone bugfix: `getWarningMessage()`, `formatStaticDate()`, `getUserTimezone()` fallback
1.6.5	2026-02-24	Timezone fixes
1.6.3	2026-02-12	Minor fixes
1.6.0	2026-02-08	Capacity limit, email notification, form overlay
1.4.1	2026-02-06	Static date formatting (4 formats, localized)
1.4.0	2026-02-06	Countdown CSS customization

Version	Date	Changes
1.3.x	2026-02-06	Countdown timer, timezone modes
1.0.0	2026-02-05	First stable version

---

15. Status Dashboard

The **Form Expiry - Status** administrator module (`mod_formexpiry_status`) shows every form that uses a Form Expiry field in one place. On installation it is placed on the **Home Dashboard** automatically (`cpanel` position). If you delete or move it, updates do not bring it back; you can create it again under **Content** → **Administrator Modules** → **New** → **Form Expiry - Status**.

What it shows

Summary tiles (by default they count the listed forms, see the settings):

Tile	Meaning
Open	Active or expiring-soon forms
Expiring soon	Forms in their warning period
Not open yet	Forms before their opening date
Closed	Expired, full or manually closed forms
Configuration errors	Only when there are any: invalid date, invalid capacity, invalid fixed timezone, or the opening date is not earlier than the expiry date

Table, one row per form:

- **Form** — link to the RSForm! Pro editor, with the next change below it: "Opens in 2 d 3 h" or "Closes in 5 h 12 min", followed by the exact time (in the administrator's timezone). "Unpublished" and "Configuration error" badges. For forms with several Form Expiry fields (or with an error) the field details can be expanded: field name, state, opening and expiry date, timezone.

- **Status** — Active / Expiring soon / Not open yet; for closed forms the reason: Expired / Full / Closed manually. With several fields, the strictest state is shown (closed > not open yet > expiring soon > active).
- **Capacity** — "27 / 30" and a coloured bar (yellow from 80%, red at 100%). With `CAPACITY_COUNT_FROM_OPEN` marked "since opening", with `CAPACITY_COUNT_CONFIRMED_ONLY` marked "confirmed only"; with a waiting list, "waiting list: 3 / 10" is shown below (1.9.0).

The module is designed for the half-width cards of the Home Dashboard; like the built-in modules, the table runs to the edges of the card. When the card is narrow (below about 440 px, e.g. on a phone), the Status column is hidden and the status appears next to the form name.

Order (default): forms with configuration errors first, then upcoming openings/closings in time order, then the other open forms, closed forms last — the most recently closed first. For closed forms, "Closed 3 d ago" is shown below the name.

The state is calculated by the same code as the frontend and the server-side validation, so the dashboard shows exactly what visitors see. Loading the data takes 5 database queries regardless of the number of forms (one more per form for forms that count from the opening date or count confirmed submissions only).

Module settings

The settings are on two tabs. The defaults are tuned for the typical use: every new form gets a closing date, and closed forms are not used again. Therefore closed forms are shown **for 7 days only** by default, after which they drop off the list automatically.

Filter tab

Setting	Default	Description
Show open forms	Yes	Forms currently accepting submissions
Show expiring-soon forms	Yes	Forms in their warning period
Show not-yet-open forms	Yes	Forms before their opening date
Closed forms	Only recently closed	`Only recently closed` / `Show all` / `Hide`
Closed within (days)	7	With "Only recently closed": how many days a closed form stays listed. Closing time: the expiry date for date-based expiry, the last submission for full forms, the time of closing for manual closure.

Setting	Default	Description
Always show configuration errors	Yes	Forms with a configuration error are listed even if their state is hidden above
Show not configured fields	No	Also list forms whose Form Expiry field has no opening date, expiry date or capacity limit
Include unpublished forms	No	Also list forms that are unpublished in RSForm! Pro
Exclude forms	—	Selected forms that never appear and are not counted in the tiles (e.g. test forms)
Ordering	Smart	`Smart` (the order described above; among closed forms the most recently closed first) / `Newest forms first` / `Title`
Maximum rows	20	Maximum number of forms in the table (`0` = no limit)

Display tab

Setting	Default	Description
Show summary tiles	Yes	Show the tiles
Tiles count	Listed forms	`Listed forms`: the tiles match the list. `All forms`: forms hidden by the filter are counted too (excluded forms never).
Show next change	Yes	Below the form name: "Opens in …" / "Closes in …", for closed forms "Closed … ago"
Show exact time	Yes	The exact date and time next to the relative time
Show capacity column	Yes	The "X / Y" column with a progress bar
Show field details	Yes	Expandable details for forms with several fields or a configuration error
Blocked attempts	Yes	**1.9.0.** Below the form name: the number of submission attempts rejected in the last 30 days

Setting	Default	Description
Close / reopen buttons	Yes	**1.9.0.** Buttons to close and reopen a form immediately (only shown with the form management permission)

If forms are hidden by the settings, a line below the table shows how many ("3 more forms are hidden by the module settings.").

Closing and reopening (1.9.0)

Open forms (active, expiring soon, not open yet) have a **Close** button below their name. After clicking it you can enter a message shown to visitors (leave it empty for the default text); the closure takes effect immediately for every visitor, and server-side validation takes it into account too. Manually closed forms have a **Reopen** button. Forms closed by date or capacity have no button — change the date or the limit in the form editor instead.

The action is sent to the server as a POST request with a Joomla CSRF token, and can only be performed from the administrator area with the RSForm! Pro form management permission (`forms.manage`).

Blocked attempts (1.9.0)

The plugin counts how many times the server rejected a submission on a form (expired, full, not open yet, closed, duplicate submission, overload). The dashboard shows the total of the last 30 days below the form name. Since JavaScript already disables the buttons, it is typically only browsers without JavaScript, bots and hand-crafted requests that reach the server — so an unusually high number points to attempted abuse.

The log stores daily counters (`#__formexpiry_blocked` : form + reason + day + count) and **no personal data** (no IP address, user or submitted data). The scheduled task (section 16) deletes rows older than one year. A breakdown by reason is available with SQL:

```
SELECT day, reason, attempts FROM `#__formexpiry_blocked` WHERE form_id = 12 ORDER BY day
DESC;
-- reason: date | capacity | not_open | dynamic | duplicate | busy | config | error
```

Permissions

The module is only shown to users who have **manage permission** (`core.manage`) for the RSForm! Pro component. The Close / Reopen buttons additionally require RSForm! Pro's **form management** permission (`forms.manage`).

---

16. Scheduled Notifications

The **Task - Form Expiry notifications** plugin (`plg_task_formexpiry`) uses Joomla's Scheduled Tasks (**Joomla 4.1+**) to send the admin email notifications — even if nobody opens the expired or full form. The

installer enables the plugin, but you have to create the task yourself:

1. **System → Scheduled Tasks → New**
2. Choose **Form Expiry – admin notifications**
3. Schedule: e.g. **every 15 minutes** (or hourly if it is not urgent)
4. Save

The task has one setting:

Setting	Default	Description
Only events from the last (days)	7	Notifications are only sent for forms that closed within this many days, so the first run does not send an email about every long-closed, no longer used form. `0` = no limit.

The task checks every Form Expiry field on every published form and sends an email for expired (date) or full (capacity) events, using the email settings of the **System - Form Expiry** plugin (recipients, subject, body, notification type — see section 9). Deduplication is shared with the frontend: an event is sent only once, even if the task and a visitor detect it at the same time. The result of each run appears in the task log ("Checked N Form Expiry field(s), sent M notification(s), skipped K older than 7 day(s), purged P old blocked-attempt log row(s)."). From 1.9.0 the task also deletes rows older than one year from the blocked attempts log.

> **Note:** by default Joomla uses a "lazy" scheduler that starts due tasks when the site is visited. For precise timing, set up a server cron or web cron (System → Global Configuration → Scheduled Tasks).

---

17. Security Notes

Version 1.9.0 contains the results of a security review. What is worth knowing:

- **The server decides.** Disabled buttons, the overlay and the countdown are display only; every submission is checked and rejected by the server. Leaving out or rewriting the hidden field, or hiding the field with an RSForm! Pro condition, does not get around it (the plugin reads the fields from the database).
- **Closed on error.** If counting the submissions or the uniqueness check runs into a database error, or the capacity lock cannot be acquired, the server rejects the submission. For an invalid date / capacity the behaviour is configurable (section 13) — choose "closed" when the deadline is strict.
- **Capacity on public forms.** On a form without login anyone can fill up the places. Countermeasures: `CAPACITY_COUNT_CONFIRMED_ONLY`, `UNIQUE_FIELD`, a CAPTCHA on the form, and watching the "blocked attempts" number on the dashboard.

- **Admin actions.** Close / Reopen is only available from the administrator area, as a POST request with a CSRF token and the `forms.manage` permission. The public `com_ajax` endpoint only returns the server time.
- **Privacy.** The plugin stores no personal data about visitors: the blocked attempts log only contains daily counts. The plugin sets no cookie of its own; session data is only stored by a closure for a single visitor requested from code.
- **Updates.** The update server is reached over HTTPS; install the package only from extensions.blaho.hu.

Important Notes

1. **The field is always hidden** — visitors do not see the field itself; it only needs to be in the form for the functionality.
2. **Server-side + client-side blocking** — Even with JavaScript disabled, the `processValidation()` method blocks submissions to not-yet-open, expired, full or closed forms on the server.
3. **The Validation Message is multilingual** — Set it separately in every active Joomla language (the same applies to `PENDING_MESSAGE`).
4. **Cache** — For date-based opening and expiry the display is correct even on cached pages (the JS calculates from the timestamps and the server time). For the capacity limit and dynamic closure to show up immediately, exclude the form's page from the page cache (the server blocks the submission either way).
5. **Timezone** — In `server` and `user_browser` modes the dates are interpreted according to Joomla Global Configuration → Website Time Zone. PHP's default timezone does not matter.
6. **Server time request** — If there is a future opening or expiry, the page calls `index.php?option=com_ajax&group=system&plugin=formexpiry&format=json&action=time` once. If a security extension or firewall blocks `com_ajax`, the JS keeps working with the browser's clock.