

Deep Lists — Configuration Guide

Package: Deep Lists (List View Manager) · Namespace `deeplists`

Everything you can configure once Deep Lists is installed: who can do what, the background job, and the eight org settings.

For getting the package in, see the **Installation Guide**. For building views, see the **Filter Reference**.

1. The permission model

Deep Lists has three permission sets, and behind them five custom permissions that are checked on the server. **Hiding a button is not the control** — the server refuses the action regardless of what the screen shows.

The three permission sets

Permission set	Persona	Can
LVM Administrator	Salesforce admins, list view owners	Everything
LVM Publisher	Team leads curating for their own team	Build and publish, within their role subtree
LVM End User	Everyone receiving views	Open views published to them, pin them, hide them

Each set is **self-sufficient**. It grants everything its persona needs without relying on the subscriber's profile, so a minimum-access profile plus the permission set is a complete and supported configuration.

The five custom permissions

Custom permission	In which set	Controls
LVM_Build_Views	Administrator, Publisher	Creating, editing and cloning views
LVM_Publish_Views	Administrator, Publisher	Publishing, unpublishing, managing audiences
LVM_Publish_All_Audiences	Administrator only	Publishing to anyone . Without it, a publisher can only reach their own role and the roles beneath it
LVM_Retire_Views	Administrator only	Retiring a view, and restoring a retired one
LVM_Manage_Settings	Administrator only	Editing settings, and the maintenance job

Customising it

The shipped sets are a starting point. To vary them, **clone the set and edit the clone** — a managed package's own permission sets cannot be edited directly, and cloning survives upgrades.

- **A publisher who may also retire:** clone **LVM Publisher**, add `LVM_Retire_Views`.
- **A publisher who may reach the whole org:** clone **LVM Publisher**, add `LVM_Publish_All_Audiences`.
- **An analyst who should see usage but change nothing:** clone **LVM Administrator** and remove `LVM_Build_Views`, `LVM_Publish_Views` and `LVM_Retire_Views`.

The role-subtree restriction

Without `LVM_Publish_All_Audiences`, a publisher may only publish to people in **their own role and the roles below it**. This relies on your role hierarchy.

A publisher with no role assigned can only publish to themselves. If your org does not use roles, grant `LVM_Publish_All_Audiences` to everyone who publishes, or the feature will appear broken.

2. The maintenance job

An hourly background job that does four things. **Nothing on this list happens without it.**

Pass	What it does
Activate	Turns scheduled publishes live at their appointed time
Roll up	Refreshes Opens (30d), last-open date and unique reach on every view
Flag	Marks stale views, and audiences whose user, role or group no longer exists
Purge	Deletes usage rows older than your retention setting

Turning it on

List View Manager → Settings → Maintenance job → Schedule hourly. The badge turns **On** and shows the next run time.

This is the only way to schedule it. The job cannot be started from anonymous Apex, and it does not appear under Setup → Apex Classes → Schedule Apex, because the class is not exposed outside the package. The button also records the schedule so the console can report its status, which a manual schedule could not.

Run now

Run now performs a single immediate pass. Use it after changing the stale threshold or the retention setting when you want to see the effect without waiting for the hour.

Checking it is running

The Settings tab shows the next scheduled run. The List Views tab shows a warning banner whenever the job is off. If **Opens (30d)** is permanently empty, the job is not running.

3. Settings

List View Manager → **Settings**. Changes save as you make them.

Setting	Default	What it does
Notify users when a new view is assigned to them	On	Announces a view to people who have just gained it
Notify users when a view is retired or removed from them	On	Tells people when they <i>lose</i> a view — retired, unpublished, or audience removed
Deliver by	In-app notification	How both notifications are delivered. See §4
Default retirement grace period	14 days	Pre-fills the grace period on the Retire dialog
Let end users hide assigned views	Off	Adds a hide control to My Views
Collect usage analytics	On	When off, nothing is recorded about who opened what
Stale view threshold	90 days	A published view with no opens for this long is flagged under Needs attention
Keep usage history for	1 year	How long individual open records are kept. Minimum 60 days

The Settings tab needs `LVM_Manage_Settings`. Everyone else sees the values but cannot change them.

These are hierarchy settings

Deep Lists stores these as a Salesforce hierarchy custom setting, so **you can override any of them per profile or per user** through Setup → Custom Settings → LVM Setting → Manage. The Settings tab edits the org-wide default; overrides you create in Setup take precedence for those users.

Most orgs never need this. It is there if you want, say, analytics collection off for one contractor profile.

Two worth a decision on day one

Collect usage analytics and **Keep usage history for** together determine how much user activity data Deep Lists retains. If your organisation restricts that, set the retention window accordingly, or switch collection off — that stops all writes immediately.

The 60-day minimum exists because the Analytics tab compares the last 30 days against the previous 30. A shorter window would silently break that comparison rather than merely shortening it.

4. Notifications

Both notification switches share one delivery channel, set by **Deliver by**.

Option	What happens
In-app notification	Deep Lists sends the Salesforce bell notification itself. Nothing to configure
Platform event (build your own Flow)	Deep Lists publishes an event and sends nothing. Your Flow decides the channel and the wording
Both	Both of the above

Only newly-reached people are notified

Republishing a view to an audience that already has it notifies nobody. Editing a view's columns and publishing again does not spam everyone who already had it — only people who genuinely gained access are told.

The person who performs the publish is never notified about their own action.

The platform events

Choose *Platform event* or *Both* and Deep Lists publishes **one event per affected person**:

Event	Published when
List_View_Assigned__e	Someone gains a view
List_View_Retiring__e	A view someone has is retired — carries the retirement date and any replacement
List_View_Access_Revoked__e	Someone loses a view immediately, through unpublishing or an audience change

One event per person rather than one per publish is deliberate: your Flow receives a single user Id and acts, with no need to resolve an audience itself. Audience resolution — nested groups, role hierarchies, de-duplication — is the part you cannot reasonably reimplement, so Deep Lists hands you the answer.

These events never leave your org. They are published and delivered inside Salesforce.

The retirement event fires at the **moment of retirement**, not at the end of the grace period, so recipients have the whole grace period to switch. There is no event when a grace period expires.

See the **Flow Integration Guide** for each event's fields and a worked example.

If notifications are not arriving

Check	
Is the relevant switch on?	Assignment and retirement are separate switches
Did those people already have the view?	Only newly-reached users are notified
Is Deliver by set to <i>Platform event</i> ?	Then Deep Lists sends nothing itself — your Flow must
Is your Flow actually running?	Setup → Paused and Failed Flow Interviews . Deep Lists publishes the event and cannot see whether your Flow succeeded

5. Retirement and grace periods

Retiring is the intended way to remove a view. It keeps the history, lets you nominate a **replacement view** to point users at, and honours a **grace period** before the view disappears. A retired view can always be restored.

Default retirement grace period pre-fills the dialog; you can override it per retirement, or retire immediately.

Retire, Unpublish, and Cancel — three different things

Action	Effect
Retire	Ends the view's life gracefully, with a date, a replacement and a grace period. Reversible via Restore
Unpublish	Takes the view from everyone now and returns it to Draft
Cancel scheduled publish	Drops a <i>staged</i> change only. The live version, its audience and its status are untouched

Cancel scheduled publish appears only while a publish is pending. If you want to drop a scheduled change without disturbing anyone, that is the one to use — not Unpublish.

6. What users see, and what you control

Behaviour	Controlled by
Which views someone sees	The audience you publish to
Which records they see inside a view	Your org's sharing model — not Deep Lists

Behaviour	Controlled by
Which fields they see	Field-level security — not Deep Lists
Whether they can hide a view	Let end users hide assigned views
Whether they can pin a view	Always on. Pinning is personal

Deep Lists cannot widen anyone's access. A view runs as the person opening it, so someone with narrower record access sees fewer rows. Hiding and pinning are personal and never change the audience — hiding a view does not remove anyone else's access to it.

7. Configuration troubleshooting

Symptom	Cause and fix
Settings tab is read-only	The user lacks <code>LVM_Manage_Settings</code>
Retire / Restore missing from the row menu	The user lacks <code>LVM_Retire_Views</code>
"You can only publish to people in your own role..."	The publisher lacks <code>LVM_Publish_All_Audiences</code> and picked someone outside their role subtree. Narrow the audience, or grant the permission
A publisher can only publish to themselves	They have no role assigned. Assign a role, or grant <code>LVM_Publish_All_Audiences</code>
Opens (30d) always empty, stale flags never appear	The maintenance job is not scheduled (\$2)
A scheduled publish never went live	Same cause. The job activates due publishes on its next run
Hide control missing from My Views	Let end users hide assigned views is off
An audience row shows "Needs review"	The stored Id no longer resolves to a real user, role or group. Remove the row and re-add the audience
I edited a view but users still see the old columns	Working as designed. Users get the published version, not your working copy. Publish again to roll it out
A view I scheduled still shows as Published	Also by design. A live view keeps its status, audience and version until the scheduled time. The row reads "Published · update <date>" — scheduling stages a change, it does not take the view down
A filter appears to do nothing	See the Filter Reference

For anything else, contact paul.tunison@gmail.com with your Org Id and the name of the affected list view.