

Deep Lists — Installation Guide

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

Getting Deep Lists installed and proving it works. About ten minutes.

Once you are through this guide, see the **Configuration Guide** for the maintenance job, notifications and settings, and the **Filter Reference** for building views.

Before you start

Requirement	Detail
Salesforce edition	Enterprise, Unlimited, Performance, or Developer
Lightning Experience	Required — the package ships no Salesforce Classic UI
My Domain	Must be deployed (required for Lightning web components)
Permission to install	"Download AppExchange Packages" and "Customize Application"
Time	About 5 minutes to install, 5 to set up

No network changes are needed. Deep Lists makes no external calls, so there are no Remote Site Settings, Named Credentials or Connected Apps to configure, and nothing for your firewall or security team to allow.

Step 1 — Install the package

1. Open the Deep Lists listing on AppExchange and choose **Get It Now**, or use the installation URL supplied by your account contact.
2. Choose **Install in Production** or **Install in Sandbox**.
3. On the install screen, select **Install for Admins Only**.

Install for Admins Only is the right choice here. Deep Lists ships three permission sets that control access precisely. Granting through those is cleaner than opening it to every profile at install time, and it is easy to widen later.

4. Approve the install and wait for the confirmation email.

Verify under **Setup → Apps → Packaging → Installed Packages**. **Deep Lists** should be listed with namespace `deeplists`.

Step 2 — Assign permission sets

Nothing is visible until you do this. Deep Lists uses permission sets rather than profiles.

Go to **Setup → Users → Permission Sets**.

Permission set	Give it to	What it grants
LVM Administrator	List view owners and Salesforce admins	Everything: build, publish, retire, restore, analytics, settings
LVM Publisher	Team leads who curate views for their own team	Build and publish. No retire, no settings
LVM End User	Everyone who should <i>receive</i> list views	The My Views app, and the ability to open views published to them

To assign one:

1. Click the permission set name.
2. Click **Manage Assignments → Add Assignments**.
3. Select users and click **Assign**.

Start with LVM Administrator for yourself, then assign **LVM End User** broadly — most people in the org need only that one.

These permission sets are self-sufficient. They work on a minimum-access profile and do not depend on anything your profiles happen to grant, so you can follow Salesforce's own recommended profile model without Deep Lists breaking.

The permission model, the five custom permissions behind it, and how to customise it are covered in the **Configuration Guide**.

Step 3 — Open the apps

From the **App Launcher** (the grid icon, top left):

- **List View Manager** — the administrator console. Four tabs: List Views, Assignments, Analytics, Settings.
- **My Views** — what your end users see.

If an app does not appear, that user has no Deep Lists permission set. Back to Step 2.

Step 4 — Confirm it works

A two-minute smoke test that exercises the whole lifecycle.

1. Open **List View Manager**.
2. Click **New List View**, pick **Opportunity**, name it `Install Test`, choose **Start from scratch**.
3. In the Builder, add two or three columns and one filter. The live preview on the right should show matching records and a count.

If the preview shows every record, your filter value may not be in a format Deep Lists can read. See the **Filter Reference** — this is the single most common early stumble.

4. Click **Publish**, choose your own user as the audience, publish immediately.
5. Switch to the **My Views** app. `Install Test` should be there. Open it and confirm the records render with your columns.
6. Back in **List View Manager**, use the row menu on `Install Test` → **Retire**, then **Restore...**, to confirm the full lifecycle.

Delete the test view when you are done, or leave it — retired views keep their history.

Step 5 — Turn on the maintenance job

Do not skip this. Salesforce does not allow a managed package to schedule Apex during install, so Deep Lists ships with its background job switched off.

Open **List View Manager** → **Settings**, find **Maintenance job**, click **Schedule hourly**.

Until you do, none of this works:

- scheduled publishes never activate;
- the **Opens (30d)** column stays empty;
- stale views and broken audiences are never flagged;
- usage history is never pruned.

The List Views tab shows a warning banner while the job is off, so you can tell at a glance.

The Settings tab is the only way to schedule it. The job cannot be started from anonymous Apex or from Setup → Apex Classes → Schedule Apex, because the class is not exposed outside the package. That is deliberate: the button also records the schedule so the console can report its status.

The job and what each pass does are covered in the **Configuration Guide**.

Step 6 — Review the settings

Still on the Settings tab, check the defaults suit you — notifications, the retirement grace period, whether users may hide views, analytics collection and retention.

All eight settings are documented in the **Configuration Guide**. The defaults are sensible for most orgs; the two worth a decision on day one are **Collect usage analytics** and **Keep usage history for**, if your organisation restricts how long user activity data may be kept.

Upgrading

Deep Lists is a managed package. Upgrades install over the existing version and **preserve all data and configuration** — views, versions, audiences, usage history and settings all survive. Custom object, field and Apex API names never change between versions.

Install the upgrade in a sandbox first if your org has customisations that reference package objects, or Flows subscribed to the package's platform events.

Uninstalling

Setup → Apps → Packaging → Installed Packages → Uninstall next to Deep Lists.

Uninstalling permanently deletes all Deep Lists records: view definitions, versions, audience assignments, usage history and per-user state. Salesforce offers to export the data first — take that export if you might reinstall.

Your own data is untouched. Deep Lists only ever *read* your Accounts, Cases, Opportunities and custom objects; it never created or modified a record on any of them, and uninstalling changes nothing about them.

Troubleshooting the install

Symptom	Cause and fix
App does not appear in the App Launcher	That user has no Deep Lists permission set (Step 2)
"Insufficient privileges" opening a tab	Same cause — check the assignment
Builder shows no fields for an object	The user lacks field-level access to that object. Deep Lists only ever offers fields the user can already see
Retire / Restore missing from the row menu	The user lacks the <code>LVM_Retire_Views</code> custom permission
Settings tab is read-only	The user lacks the <code>LVM_Manage_Settings</code> custom permission
Opens (30d) is always empty	The maintenance job is not scheduled (Step 5), or has not run yet. Use Run now
A scheduled publish never went live	Same cause. The job activates due publishes on its next hourly run
A filter seems to be ignored	See the Filter Reference — most often a date typed without a time, or commas instead of semicolons

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