Drupal 10 End of Life — Client Upgrade Checklist

Working checklist for freelancers and small shops. Companion to the post *Drupal 10 Ends December 9: The Freelancer's Client Upgrade Checklist*.

Drupal 10 reaches end of life on December 9, 2026. 16 calendar days left — about 12 working days.

> After that date the Drupal Security Team stops publishing fixes for any Drupal 10 release. Drupal 11 minors have their own support windows, so they need checking too.

Site / client: _____ Reviewed: _____

1 Inventory every client site and its exact core version

Before you quote anything, find out what each client is actually running. A spreadsheet is fine.

- [ ] Exact core version — from `composer.json`, `composer.lock`, or `/admin/reports/status`. "Drupal 10" is not specific enough.
- [ ] PHP version on production *and* on any staging environment.
- [ ] Count of contrib and custom modules — note custom modules and themes especially, whoever wrote them.
- [ ] Hosting platform, and whether the host publishes its own end-of-life guidance.
- [ ] Contractual standing: retainer, ad hoc, or no current agreement.
- [ ] Sort into three buckets: already on a supported Drupal 11 minor / on Drupal 10 and close to ready / on Drupal 10 with real blockers.

> Anything still on Drupal 7, 8 or 9 is a separate conversation — those are already unsupported. Drupal 7 needs a migration, not an upgrade.

2 This week: patch every Drupal 10 site to the latest release

Not optional, and independent of the upgrade. An unpatched Drupal 10 site is exposed today, not just after December 9.

- [ ] Apply current core security releases across every Drupal 10 site.
- [ ] Run core updates through Composer on a staging copy first.
- [ ] Run `drush updatedb` and `drush config:export`, then commit the results.
- [ ] Smoke-test forms, logins, and any commerce or payment flows before deploying to production.
- [ ] Get the site to Drupal 10.3+ on PHP 8.3 — needed regardless, so bill it as security maintenance *and* upgrade groundwork.

3 Audit upgrade readiness with Upgrade Status and Composer

A 10 → 11 upgrade is mostly work done *on* Drupal 10. Fix deprecations and incompatible dependencies first; switch core last.

- [ ] `composer require drupal/upgrade_status`, then run its report — deprecated API usage in custom code, plus contrib without a Drupal 11-compatible release.
- [ ] Run `composer why-not drupal/core 11` to see which dependencies are in the way.
- [ ] Confirm the PHP path to 8.3 before the switch.
- [ ] Log every blocker as a line item: contrib update available / contrib patch needed / custom code fix / module replacement or removal.

4 Ship the fixes, then switch core

Release every fix while the site is still on Drupal 10, and only then switch core. Small deploys, easy rollback, visible progress.

- [ ] Deploy each readiness fix to production on Drupal 10, one small release at a time.
- [ ] Readiness report clean before touching core.
- [ ] Bump `drupal/core-recommended`, `drupal/core-composer-scaffold` and related constraints to `^11`, then run the update.
- [ ] Run `drush updatedb` and a full config export.

> If the deadline slips, the site is already most of the way to Drupal 11 and still running patched Drupal 10.

5 Be honest about timelines and set expectations

A mid-size site runs 11–24 working days. Do the math against the date before you promise anything.

- [ ] Small, clean sites: aim to complete the upgrade before December 9.
- [ ] Mid-size or complex sites: keep on latest patched Drupal 10, start readiness work now, schedule the core switch once blockers clear.
- [ ] Tell those clients plainly that no new Drupal 10 security fixes will be published after December 9.
- [ ] Have an answer ready for "can't we just wait for Drupal 12?" — waiting means running unsupported software in the meantime.
- [ ] Public sector and regulated clients: send the guidance written for agencies.
- [ ] Client won't fund it? Put the risk in writing — a short dated email confirming you told them protects both of you.

6 Don't forget your Drupal 11 clients

"Already on Drupal 11" doesn't mean done. Drupal 11 minors have their own end-of-support dates.

- [ ] Check each Drupal 11 site's minor against the core release schedule.
- [ ] Move anything on an older minor to a supported one — same maintenance window as the Drupal 10 patching. One pass, every client covered.

★ Today

The short list, if you do nothing else.

- [ ] Build the client inventory spreadsheet.
- [ ] Patch every Drupal 10 site to the latest security release.
- [ ] Check Drupal 11 minors against the release schedule.
- [ ] Run `upgrade_status` on your riskiest sites.
- [ ] Send each client a dated note with their status and their options.

Patch first. Tell every client. Upgrade on a schedule you can keep.

From drupalhelps.com — *Drupal 10 Ends December 9: The Freelancer's Client Upgrade Checklist.*

Ticks are saved in this browser only, per site name. Nothing is uploaded.