Step-by-Step Guide to Migrating from Drupal 9 to Drupal 10 or 11

Like any digital platform, Drupal also evolves over time, providing new updates, better security, and better tools for users and developers alike. If your website is still running on Drupal 9 or earlier, it's probably missing out on a whole new world of features and performance improvements that are already possible with Drupal 10 and Drupal 11.

With Drupal 11 set to improve integration with modern systems, streamline admin workflows, and improve the content editing experience, updating your CMS is a smart move in 2025. In addition to keeping up with digital expectations, maintaining top-notch security, and maximising performance and scalability, it also unlocks better scalability and performance.

Whether you’re planning to do it in-house or thinking about using professional Drupal migration services, this guide will provide you with everything you need to know. Even if your site isn’t on an older version, upgrading to the latest release keeps you ahead of potential issues and lets you make the most of what Drupal has to offer.

Now, let’s walk through the step-by-step Drupal migration process so you can make informed, confident decisions for your website’s future.

Step-by-Step Migration Guide: Upgrading from Drupal 9 to Drupal 10 or 11

You may feel concerned about migrating from Drupal 9 to the latest version, but the process is much easier than going from Drupal 7 to 9, as it is much more streamlined. A well-thought-out migration strategy can minimize downtime and future-proof your platform. Let's go through a clean, optimized migration.

Get Familiar With What’s New

Before making any changes, you need to consult the release documentation for Drupal 10 or 11. The release notes will have important information like what’s new, what’s been removed, and what’s been added. Knowing this ahead of time will save you some unplanned surprises.

Secure a Full Backup of Your Existing Site

Never start an upgrade without complete safety. You need to back up everything: your database, files, custom modules, themes, and settings. If anything breaks during the upgrade process, your safety net will allow you to restore your site quickly to its original stable state.

Ensure Your Drupal 9 Version Is Ready

It is recommended that you already use one of the final minor versions of Drupal 9-preferably 9.4.x or higher. Earlier versions will not support a direct upgrade path to Drupal 10 or 11.

Ensure that your host environment meets the technical requirements for either Drupal 10 or 11 (PHP 8.1+).

Review and replace any deprecated themes or modules you may be using. For example, CKEditor 4 over CKEditor 5, or the old Seven theme, to a modern theme such as Olivero.

Audit and Update Your Codebase

Now is the time to clean your custom code. Use tools like Drupal Rector to scan for outdated functions and prepare your modules and themes for the newer code standards. While you're at it:

Replace unsupported tools (like Drupal Console).

Use Upgrade Status to check which contributed modules can be used with Drupal 10 or 11.

Make sure to update your modules with Composer and edit composer.json to include support for the new majors.

When you have cleared all the modules and dependencies, uninstall and remove Upgrade Status from your site.

Handle Any Default File Changes

A Drupal upgrade may overwrite files such as robots.txt and .htaccess. If you made custom modifications to these files, be sure to keep a record of them so you can reapply them once the new version is installed.

Prepare Your Codebase for Core Upgrade

It's important to prepare your site's backend before upgrading to Drupal 10 or 11. It's a good idea to run a test update first to avoid surprises. This involves adjusting permissions, updating packages, and checking for conflicts. Once everything checks out, you can do the upgrade and update the database.

- Adjust permissions
- Add the required core packages via Composer
- Run a dry run to check for conflicts

If successful, proceed with:

- composer update
- composer install
- Then update your database
- Reapply read-only permissions

Final Checks After the Upgrade

When the upgrade is done, you now need to stabilise and tune your site to power everything that has been learned in this upgrade process:

- Export your configuration to store your updated configuration that will occur during the upgrade.

- The final steps would be to push your changes into version control (Git) and finally, use git diff to ensure everything is in sync.

- Use PHPCS (PHP CodeSniffer) for your custom code to make sure you are adhering to the latest coding standards prescribed by Drupal.

Run your site's test suite, e.g., PHPUnit, Behat, and verify that there have not been any breakages in any existing functionality during the upgrade.

Monitor logs in real time to catch any emerging issues:

- bash
- CopyEdit
- Drush watchdog:tail

Don’t Forget User Testing

Test forms, navigation paths, login flows, and content displays manually in addition to automated tests. This step will help you detect any user-facing bugs that automated scripts may miss.

You're Ready to Go Live!

With testing complete and adjustments made, you're ready to launch your new Drupal 10 or 11 site. Make sure to:

- Notify all stakeholders.
- Pay attention to how the site performs in the early days following the launch.
- Have your development team on standby for post-launch adjustments.

Essential Tools to Simplify Your Drupal Migration

With a wide range of reliable tools and modules designed to simplify the entire upgrade journey, migrating your Drupal site doesn't have to be complex. These tools will ensure a smooth, efficient, and error-free migration from Drupal 7, 8, or 9 to Drupal 10 or 11.

Upgrade Status
Extended Migration Modules
Other Developer-Focused Utilities
Core Migration Modules (Built-in)

Upgrade Status

It is always a good idea to check your current site for compatibility. Upgrade Status scans your existing setup and flags any modules, themes, or core components that may not work with newer Drupal versions. Moreover, it provides recommendations for updates or alternatives, making it a good place to start with your migration.

Core Migration Modules (Built-in)

Drupal core is made up of several key modules that help you easily handle site migrations:

Migrate

This is the foundation module for all of the site content and site configuration migration, as it handles all of the migration work.

Migrate Drupal

Provides migrations from previous Drupals, i.e., 6, 7, and 8, to newer versions.

Migrate Drupal UI

Provides a very simple one-page user interface (located at /upgrade) that provides a more intuitive method for providing upgrades, primarily for users coming from versions of Drupal 7.

Migrate Drupal Multilingual

Supports migrations involving multilingual content (still experimental in some versions).

Extended Migration Modules

If your migration has more complex data or external data sources, these modules extend the default migration tools:

Migrate Plus

This extends migrations' support for additional data sources, improves configuration management, and provides more flexibility in mapping fields.

Migrate Tools

This adds useful Drush commands, and a UI to run and manage migrations, a handy tool for developers or technical site owners.

Migrate Source UI

This module allows you to upload files (like a CSV or XML file) as migration sources. This is valuable to people who need to import structured data.

Migrate Scheduler / Migrate Cron

For users of Cron jobs, these modules provide an automatic migration by leveraging a Cron job to schedule when the data should be extracted.

Drupal Rector

This handy developer tool takes your custom code and automatically replaces outdated functions with modern equivalents, so that you can make custom modules or themes compatible with Drupal 10 and higher while simultaneously reducing the amount of manual work involved in updating them.

Composer

The Composer package manager performs all the necessary work for you to update Drupal modules and dependencies correctly. Upgrade Status also integrates with Composer to ensure that everything has been updated properly and efficiently.

Why Upgrading to Drupal 11 Is a Smart Move?

There is a substantial improvement in functionality, performance, and developer experience with Drupal 11. This release offers a much smoother upgrade path for organisations that still use old versions of Drupal 7, 8, and 9.

Here are some reasons why it's worth switching to Drupal 11:

Better Performance with Modern PHP and Symfony

The minimum requirements of Drupal 11 use PHP 8.1+, which is an incredible upgrade for the essential performance enhancements, memory fatigue, and processing speed. Additionally, with Symfony 6, developers benefit from security improvements, enhanced debugging capabilities, and a clean and solid code base.

Modern Front-End Development with JavaScript and SDCs

By reducing reliance on jQuery and adopting ES6+ syntax, Drupal 11 continues its progression towards modern JavaScript standards. Single-Directory Components (SDCs) help developers manage UI elements more effectively by gathering HTML, CSS, and JS files under one directory.

Simplified Site Building

This new feature allows site builders to quickly apply great pre-configured functionality. If you want to add a blog, event calendar, or contact form, Recipes provides you with reusable configurations that will save you time and make it easier.

Improved Content Authoring and Layout Control

Experience Builder and CKEditor 5 allow content teams to easily build engaging pages. CKEditor 5 introduces collaborative editing and a cleaner interface, and Experience Builder allows teams to develop beautifully laid out pages from structured content with very little effort.

A Fresh Look for Sites and Admins Alike

Drupal 11 introduces modern, accessible themes:

- Olivero, the new default front-end theme, replaces Bartik; it features a clean design that is responsive and meets WCAG AA-level accessibility standards.
- Claro, the new admin theme, adds modern usability, keyboard navigation, and a mobile-first approach to the user interface.

Challenges Faced When Migrating from Drupal 7,8 to 10 or 11

UNCLEAR PROJECT SCOPE
POOR MODULE STRATEGY
STRENGTHENING SITE SECURITY
IDS LOGIC
SKIPPING A FULL SITE ASSESSMENT
UNDERESTIMATING COMPLEX COTENT STRUCTURES
MANAGING THE LEARNING CURVE
REWORKING CUSTOM MODULES AND THEMES

It is a major upgrade to move from Drupal 7 or 8 to Drupal 10 or 11, and while it opens the door to a faster, more flexible platform, it is not without its challenges. Here are eight major ones, with suggestions about how to overcome them.

Unclear Project Scope

It is common for teams to run into last-minute issues when they fail to define what's within and outside the scope, including SEO redirects, backups, user training, and testing. Setting a clear, detailed scope upfront ensures all functional and business-critical areas are covered before the start of the project.

Skipping a Full Site Assessment

There's a risk of delays, overruns, and missing key features post-launch if you don't audit your Drupal 7 or 8 site first. If you don't understand your existing modules, content types, custom code, and integrations, you run the risk of delays, cost overruns, and missing key features. An accurate technical assessment will help you upgrade more smoothly.

Underestimating Complex Content Structures

A Drupal 7 site usually has deeply nested or referenced content structures, which cannot be migrated cleanly by default. When ignored, this complexity can lead to broken links, missing data, and poor user experiences. A carefully mapped content type and relationship in advance helps prevent downstream issues.

Poor Module Strategy

Not every Drupal 7 or 8 module has a simple match in Drupal 10 or 11. Some may have been deprecated, replaced, or reorganised completely. Underestimating the complexity of a simple switch will lead to problems for you. You need to check your module list, find a supported replacement for each, and then ensure the replacements will be compatible with your website before the upgrade.

Managing the Learning Curve

Drupal's newer versions have numerous new concepts. Developers and content editors will require time to adapt to the new content management workflows in Drupal. Training resources are available for developers, content editors, and website administrators - online courses, official documentation, and testing or development sandbox environments all help acclimate your team before the final switchover.

Reworking Custom Modules and Themes

Custom modules and themes built for Drupal 7 won't work out of the box on newer versions. These often need to be rewritten or undergo some heavy mods to work in the coding standards and API for Drupal 10 or 11.

Strengthening Site Security

Security can’t be compromised. Keep all modules updated, enforce proper access controls, and perform security audits. Ensuring a clean and secure build of your Drupal 10 or 11 site protects both your data and your users.

Final Words: Future-Proof Your Drupal Site with Confidence

A Drupal site upgrade isn't just a technical requirement; it's also an opportunity for your audience to receive modern, accessible experiences. Whether you're moving from Drupal 9 or making the leap from an earlier version, this process allows you to restructure your site architecture, implement new features, and stay on top of the latest web standards.

But the success of any migration to Drupal depends on a thoughtful plan, deep platform knowledge, and a steady hand. Even though some upgrades may appear straightforward, they are characterized by hidden complexities such as custom modules, integrations, and accessibility needs that require professional assistance.

Need Expert Help? Partner with IDS Logic
IDS Logic's Drupal experts will guide you every step of the way, from planning and development to launch and post-go-live support.

Let’s take your website to the next version—with zero disruption and full confidence.

Talk to our Drupal experts today.