

Drupal CLI in core **finally landed in 11.4**, how and why we got here

Drupal 11.4 ships `dr`, an experimental CLI in core. Why it matters, why it took ten years, and what changes for Drush workflows →

Drupal has depended on an external tool for its daily operations

- **Drush is the de facto CLI:** cache rebuilds, database updates, config import and export, deployments. It lives **outside core**.
- **Core's own script was limited:** `core/scripts/drupal` (2018) was not extensible and shipped only 3 commands.
- **Version sync broke updates:** every Symfony upgrade in core forced a new Drush release. Sites waited for Drush to catch up.
- **No CLI out of the box:** a usable Drupal CLI appears only after running `composer require drush/drush`.

Rebuild Drush in core, or keep promoting it?

PROMOTE DRUSH

"Instead of reinventing the wheel we could promote Drush instead or even include it in our composer.json as a requirement." - hchonov, 2019

A CLI IN CORE

A core command **only supports the version it ships in**. No two-way coupling with Drush, and commands follow the **core process**: review, tests, BC policy.

Ten years of the same debates

"attempts stalled due to debates about command name or bootstrap details or numerous other bikeshed details."
- Moshe Weitzman

Attempts in **2014** and **2019** ended in the same inertia. Each position was reasonable at the time:

- **Overlap:** core did not want two approaches.
- **Timing:** deferred to 8.1.x, then pushed minor to minor.
- **Name:** `drupal` was taken by Drupal Console.
- **Bootstrap and discovery:** no design ever converged.
- **Ownership:** **nobody to drive it.**

In 2026, every old blocker had an answer

- **Drush:** all its commands were converted to vanilla Symfony Console, so any Symfony application can discover them.
- **Bootstrap:** a short `Application::find()` override (about a dozen lines, used by Drush since Drush 9) lets booted and non-booted commands share one script.
- **Name:** Drupal Console was sunset in April 2025, freeing the `drupal` binary. The name decision moved to its own policy issue: result is `dr`.
- **Discovery:** standard Symfony. A compiler pass picks up `#[AsCommand]` classes in `src/Command/`.
- **Ownership:** a **CLI in Core community initiative** with the Drush lead, core committers and a roadmap.

The Drush lead wants Drush gone, and core agrees

"My end goal here is the death of Drush. It has served an incredibly long and useful life." - Moshe Weitzman, Drush lead maintainer

"There is agreement that a system-wide CLI would be a good thing for Drupal." - Dries Buytaert (summarising the **core committers'** discussion, March 2026.)

Each disagreement was settled on its own

- **Depend on Drush first?** The temporary dependency risked a half-way state with two-way coupling. It was dropped for **one command at a time**.
- **The name:** `dex`, `drupal` or `dr` moved to its own policy issue so it would not block the code. Result: `dr`, like `wp`, `ng` or `gh`.
- **11.x or Drupal 12?** Shipping in 11.x lets sites migrate slowly with Drush as a fallback.



What dr ships in Drupal 11.4

Run it with `vendor/bin/dr`, or `ddev exec dr` in DDEV. It **discovers Symfony Console commands** from core and contrib.

```
drush cr      → dr cr
drush cron    → dr cron
drush status  → dr status
drush si      → dr install
drush recipe  → dr recipe
```

Also: `content:export`, `recipe:info`,
`quick-start`, `server`.

STATUS

@internal

Experimental.

The roadmap to stable is still open.

Drush keeps working during the transition.

Two options for contrib commands

The [official porting guide](#) covers the details.

PLAIN SYMFONY COMMAND

Discovered by **both** Drush and `dr`. Recent Drush commands need little or no work.

SERVICE WRAPPER

Logic in a Drupal service, **two thin wrappers**. For commands that rely on Drush formatters.

Remote site management will likely need a new workflow

`dr` works against the **local Drupal instance**. It has no site aliases, and the roadmap lists no remote command. Remote commands depend on aliases, so **they will most likely not be covered**.

"Along the way, we may lose some niceties like Drush config and Site aliases." - Moshe Weitzman

`drush sql:sync` and `sql:dump` are directly affected. How remote management gets solved is still unclear, and probably worth another article.

Deployments still need Drush

`updatedb`, `config:import` and `deploy` are **not in** `dr` **yet**. Several commands are fixed only in dev branches, not in 11.4.

Keep Drush in deployment scripts for now, and use `dr` **alongside it**.

Essential commands are targeted for Drupal 12.0.0. A feature-complete set is expected in a minor release, most likely in 2027.

4/14

roadmap
commands
available in 11.4.0,
`site:install`
with limits

The transition has started

A CLI in core removes a **decade-old dependency** on an external tool. This time the conditions are in place: Drush is ready to step aside, core is aligned and the code has shipped.

Knowing it now avoids being wrong-footed, especially on **remote operations**. Workflows built on **aliases** and `sql:sync` will likely need more than replacing `drush` with `dr`.

What lands next depends on usage and feedback.



Ready to try dr on your modules?

Port your contrib commands, explore `dr` on real projects and share what is missing in **#cli-in-core**.
Drupal moves forward with community help.

Follow us for more on Drupal and web development.



Save



Share



Comment

