exp:install for hosts

The console install as a host uses it from scripts and panels: the options that matter, the database password from the environment, the administrator password rules, dry run and print, exit status, and examples for each…

Source file: doc/eng-us/help/hosting/provisioning/exp-install.md Last changed 2026-11-04
https://doc.exponential.earth/learninminutes/exphelp/viewer/help/hosting/provisioning/exp-install.md

This page is for system administrators and developers who build a host's provisioning on the console install, `exp:install` (`bin/php/install.php`). It answers: which options a host sets, how secrets are passed, what the command does with passwords, how to test before installing, and how a script knows it worked. The reference is Chapter 7.

Contents

1. What it is
2. The options a host sets
3. Secrets
4. The administrator password
5. Test first: print and dry run
6. Exit status
7. Examples by database

What it is

`exp:install` "is a front end to the kickstarter. It turns its options into a temporary `kickstart.ini`, runs `expKickstarter` in the same process, removes the temporary file and puts back any `kickstart.ini` that was there before" (4.2). Every option has a default, so `php bin/php/console exp:install --apply` alone installs the `sevenx_multisite` site package on SQLite with the administrator `admin` and a generated password.

The options a host sets

Option	Default	Set it to
`--db=`	`sqlite`	`sqlite`, `pgsql`, `mongodb`, `mysql` (also `mariadb`) or `oracle`
`--db-host=`, `--db-port=`, `--db-name=`, `--db-user=`	per engine	the database you created for this customer; never the `root` or `postgres` defaults on a real site
`--db-action=`	`remove`	`remove` empties the database first; on a new, empty database `remove` and `ignore` give the same result
`--url=`	`http://localhost`	the customer's address; "every siteaccess's address is built from it"

Option	Default	Set it to
`--access=`	`url`	`url` (`/admin`), `host` (with `--host` and `--admin-host`) or `port`
`--package=`	`sevenx_multisite`	the site package of the plan (Site packages)
`--language=`, `--languages=`	`eng-US`	the customer's languages
`--title=`	`Exponential`	the site name
`--email=`	`nospam@exponential.earth`	a real address of the customer; "it becomes `[MailSettings] AdminEmail`"
`--password=` / `--random-password`	generated	see below
`--force`	off	only for a deliberate reinstall

Full list with every check and message: 7.3 Options.

> **Note**
>
> When `extension/exp_adminui` is in the installation, an `adminui` siteaccess is made too, reached by `/adminui` whatever `--access` says (7.3).

Secrets

- **Database password**: pass it in `EXP_INSTALL_DB_PASSWORD`, not `--db-password`. An option "is visible to every user of the machine in the process list (`ps`) while the installation runs, and it stays in the shell history. The variable is visible to neither."
- **Administrator password**: from your secret store with `--password=`, or let the command generate one and read it from `var/log/initial-admin-password` (mode `0600`).
- **The saved configuration** in `var/log/exp-install-<date>.ini` has passwords masked.

The administrator password

From 7.5: `exp:install` "never installs a password that is empty, short or well known".

You give	Result
no `--password`	a generated password
`--random-password`	a generated password, also when `--password` is given
`--password=` with at least 10 characters (`[UserSettings] MinPasswordLength`) that is not well known	your password
a shorter or well-known password	a generated password; the summary says why
an empty `--password=`	refused, exit status 1

A generated password has 24 characters from a secure random source. It is shown once in the summary and written to `var/log/initial-admin-password`. Deliver it to the customer through a secure channel and delete the file.

Test first: print and dry run

Mode	Does
`--print`	checks the options and prints the configuration, passwords masked; no database, no network
`--dry-run`	checks the options, connects to the database, fetches the packages into a temporary repository; writes only logs

A dry run "also runs on a directory that already holds an installation, without `--force`" and ends with "Dry-run completed: database and remote packages verified. Stopped after SiteDetails, nothing written."

Exit status

`0` when installed, or for a passing `--dry-run`, `--print` or `--help`; `1` for everything else, including "the directory already holds an installation and neither `--force` nor `--dry-run` was given". A script can rely on it:

Shell

```
php bin/php/console exp:install --dry-run --db=pgsql --db-user=<db user> --url=https://www.exampl
e.com \
  && php bin/php/console exp:install --apply --db=pgsql --db-user=<db user> --url=https://www.exa
mple.com \
  || echo "installation failed, see var/log/setup.log"
```

When the run fails, "maintenance mode stays on so that visitors do not see a half-installed site".

Examples by database

In the official order. Run each as the site's user, with `EXP_INSTALL_DB_PASSWORD` set where a server is used.

Shell

```
# 1. SQLite 3 (the default): no database server
php bin/php/console exp:install --apply --url=https://www.example.com --email=<contact>

# 2. PostgreSQL
php bin/php/console exp:install --apply --db=pgsql --db-host=<db host> --db-name=<db> --db-user=<
db user> \
  --url=https://www.example.com --email=<contact>

# 3. MongoDB (a database user is required)
php bin/php/console exp:install --apply --db=mongodb --db-host=<db host> --db-name=<db> --db-user
=<db user> \
  --url=https://www.example.com --email=<contact>

# 4. MariaDB and 6. MySQL (same driver)
php bin/php/console exp:install --apply --db=mysql --db-host=<db host> --db-name=<db> --db-user=<
db user> \
  --url=https://www.example.com --email=<contact>

# 5. Oracle (service name becomes host:port/service)
php bin/php/console exp:install --apply --db=oracle --db-host=<db host> --db-name=<service> --db-
user=<db user> \
  --url=https://www.example.com --email=<contact>
```

Engine-specific notes are in 7.9 and on the database pages of this section, from SQLite for hosts to MySQL for hosts.

Provisioning a public demo. `EXP_DEMO_MODE=1 ./console exp:install --apply --password=<the published password> ...` installs the published administrator password as given, also a well-known or short one (`publish`, `publishing`); without the variable it is replaced by a generated one in `var/log/initial-admin-password`, which a demo build should treat as a failure. Never provision a customer's site with the variable. Recipe: run a public demo on LiteSpeed and Plesk.

Related pages

- One-click installers
- Automating provisioning
- Install in one command
- Free documentation: this page is free to copy, translate and improve under the GNU FDL 1.3.

> **Become an Exponential Partner**
>
> Partners share their information, provide funding, contribute back and develop the products with us, and that is how Exponential grows. A partner website and dashboard are planned for mid-2027: please become a partner today to speed up our growth rate and land new features faster.
>
> Partners: start here | How to apply

Exponential Copyright © 1998-2026 Exponential Foundation and 7x
Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".
This documentation is free documentation, licensed under the GNU Free Documentation License 1.3 or later:
https://www.gnu.org/licenses/fdl-1.3.html.