Importing content

For CMS developers loading content into a site, the import tools that exist and which one fits which source. It covers xrowextract, the kernel's ezcsvimport, packages, the ZIP importer, office documents, RSS import...

Source file: `doc/eng-us/help/cms-developers/importing-content.md` Last changed 2026-11-04
https://doc.exponential.earth/exphelp/viewer/help/cms-developers/importing-content.md

This page is for developers who have to fill a site with content from somewhere else: a spreadsheet, another CMS, an old Exponential site, a folder of photos. Use the tools that exist first and write a script only for what they cannot do. Every tool below is described in depth in Exponential's import and export tools; this page helps you choose.

Contents

1. Which tool for which source
2. xrowextract: the main importer
3. ezcsvimport: the kernel's simple importer
4. Packages: between Exponential installations
5. Media, documents and feeds
6. Layouts
7. Over HTTP
8. A script of your own
9. Rules for every import

Which tool for which source

Source	First choice	Notes
A spreadsheet or CSV of one kind of item	xrowextract with `--map`	dry run by default
An export of another CMS (WordPress, Drupal, Joomla)	convert to CSV, XML or JSON, then xrowextract	no tool reads those exports directly
Another Exponential site	packages, or a copy of the database when the whole site moves	
A folder of images and files	the ZIP importer of the upload page	keeps the folder structure if asked
Office documents	`ezodf`	needs LibreOffice or OpenOffice on the server
A news feed during a transition	RSS import	only the feed's newest items
Layouts and their rules	`exp:layouts export` and `import`	between Exponential installations
A system reachable only over HTTP	a script of your own with the kernel API	

The full table: Which tool for which job.

xrowextract: the main importer

`xrowextract` (`se7enxweb/xrowextract`, required by the root `composer.json`) reads CSV, XML or JSON and creates or updates objects of one class below a parent node. It writes nothing without `--apply`, can match existing objects for updates, maps foreign columns to attributes, runs large files in the background and resumes after an interruption:

Shell

```
./console ext:xrowextract:import --file=var/tmp/import/events.csv --class=exp_event --parent=<nod
e id> \
    --map=title=title,start=start_date
./console ext:xrowextract:import --file=var/tmp/import/events.csv --class=exp_event --parent=<nod
e id> \
    --map=title=title,start=start_date --apply
```

The easiest way to get a format right: export one sample object of the target class with xrowextract and copy its columns (xrowextract, Importing with xrowextract).

ezcsvimport: the kernel's simple importer

Shell

```
./console exp:ezcsvimport --class=exp_event --creator=<user object id> --storage-dir=var/tmp/impo
rt/files <node id> var/tmp/import/events.csv
```

The file uses `;` as separator and has no header row: column *n* goes to the class's *n*-th attribute through `fromString()`. It only creates, so a second run makes duplicates; use it for a first load of a simple class.

Packages: between Exponential installations

Packages (`.ezpkg`) carry classes and content objects with their files from one Exponential installation to another, and are the way to ship classes with your project (Packages, Site packages and installers). They are not a format other systems produce.

Media, documents and feeds

- **ZIP importer** (`ezmultiupload`): each file becomes an image or file object by type; *Keep the folder structure* makes folders. Limits are in `ezmultiupload.ini [ZipSettings]`. Images whose class requires an alternative text are skipped, because a ZIP has none (Media and uploads).
- **Office documents** (`ezodf`): Writer documents to content objects and back.
- **RSS import** (**Setup** > **RSS**): new feed items become objects of a chosen class; the `rssimport.php` cronjob runs it (RSS feeds).

Layouts

Shell

```
./console exp:layouts export --all --output=var/tmp/layouts.zip
./console exp:layouts import var/tmp/layouts.zip --dry-run
```

Layouts, shared layouts and their mapping rules travel as one ZIP file (Export and import layouts).

Over HTTP

The REST interface can create and remove nodes at v2 (updates answer 501, not implemented yet), and `expservices` offers the administration's functions as HTTP services (The REST API, Remote services and apps). Use them when the importer must run on another machine; for large imports a script on the server is faster.

A script of your own

When no tool fits, write a script with the kernel API: read the source, map each item, then create or update through `eZContentFunctions` with a remote ID per item, so runs can be repeated (Content from PHP, and the skeleton in A custom import script).

Rules for every import

1. **Model first.** Create the classes and the folder tree before importing (Designing a content model).
2. **Remote IDs from the source** (`wp-1234`, `crm-42`) on every object: updates instead of duplicates, and relations resolvable in a second pass.
3. **Media first**, then containers in parent order, then items, then relations and inner links.
4. **Dry run, then a copy, then live.** Keep the source files out of the web root (`var/tmp/` is fine on a development machine).
5. **Afterwards**: rebuild the search index (`php bin/php/updatesearchindex.php --clean`), check URL aliases, clear the content caches (Search, URL aliases and routing).
6. **No personal data you do not need.** Import user accounts only with a legal basis, and test with addresses at `example.invalid`.

Related pages

- Mapping content to Exponential
- Migration checklist
- Choosing the right import or export tool
- 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.