Mapping content to Exponential

How the concepts of any other system map to Exponential's content model - classes and datatypes, objects, versions and translations, locations, sections and states, users and passwords, media, taxonomy, comments and...

Source file: `doc/eng-us/install/migration/content-model-mapping.md` Last changed 2026-11-04
https://doc.exponential.earth/learninminutes/exphelp/viewer/install/migration/content-model-mapping.md

Whatever the source, a migration ends in the same building blocks. This page explains each one in the terms of a migration and lists the decisions to make. The path guides refer to it for the details.

Migration hub · Checklist · Import and export tools

Contents

- 1. The building blocks at a glance
- 2. Content classes and datatypes
- 3. Objects, versions and translations
- 4. Locations: the content tree
- 5. Sections and object states
- 6. Users, groups, roles and passwords
- 7. Media
- 8. Taxonomy
- 9. Comments, forms, products
- 10. URLs
- 11. Search engine metadata
- 12. Dates and authors

1. The building blocks at a glance

In many systems	In Exponential	Note
Content type, post type, entity bundle, model	**content class**	defined in **Setup > Classes**, or carried in a package
Field, custom field, property	**attribute** of a **datatype**	each class attribute has an identifier
Entry, post, page, node, item	**content object**	with versions and translations
Category tree, menu hierarchy, folder	**locations** (nodes) in the content tree	an object can have several locations; one is the main one
Category, tag, term	locations, `ezkeyword`, or `eztags`	see 8
Asset, attachment, media item	an **image**, **file** or **media** object	reusable, related or embedded
Language, locale	**translation** of the same object	languages are set up first

In many systems	In Exponential	Note
Status (draft, published)	version status; **object states** for workflow states	
Site area with its own access	**section**	roles can be limited to sections
Slug, permalink	**URL alias**	made from the name, or set by hand
Redirect	URL alias with redirect, or **URL wildcard**	
SEO fields	`xrowmetadata` attribute, or plain attributes	
User, member, customer	**user object** (class `user`) in user groups	
Role, capability, permission	**role** with **policies**, assigned to users or groups	

2. Content classes and datatypes

Make one class per source type that has its own fields. Keep the class identifiers short and stable: templates, settings and imports refer to them. The kernel's datatypes (`kernel/classes/datatypes/`) and the extension datatypes of the default distribution cover most fields:

Source field	Datatype	Note
Single-line text, title	`ezstring`	the object name usually comes from it (the class's name pattern)
Plain multi-line text	`eztext`	
Rich text, HTML body	`ezxmltext`	HTML is converted to XmlText; scripts, styles and inline event handlers do not survive, by design
Number	`ezinteger`, `ezfloat`	
Yes/no	`ezboolean`	
Date, date and time, time	`ezdate`, `ezdatetime`, `eztime`	
E-mail, URL	`ezemail`, `ezurl`	
Single or multiple choice	`ezselection`, or `enhancedselection2`	
Image	`ezimage` (in an image class, or directly on the item)	
File download	`ezbinaryfile`	
Video, audio	`ezmedia`	or a link to the hosting service
Reference to other content	`ezobjectrelation`, `ezobjectrelationlist`	resolve by remote id after all items exist
Keywords	`ezkeyword`	

Source field	Datatype	Note
Tags with a hierarchy	`eztags` (`se7enxweb/eztags`)	
Map location	`ezgmaplocation` (`se7enxweb/ezgmaplocation`)	
Rating	`ezsrrating` of `ezstarrating` (`se7enxweb/ezstarrating`)	
Table of values	`ezmatrix`	
Author list	`ezauthor`	
Country	`ezcountry`	
Price	`ezprice`, `ezmultiprice`	shop classes
A page made of blocks	Exponential Layouts (`explayouts`), or `ezflow`	layouts are separate from objects; see the guides
Link with internal or external target	`exp_enhanced_link` (`se7enxweb/exp_enhanced_link`)	
Repeating groups of fields	child objects of a small class, or `ezmatrix`	there is no nested "repeater" datatype
Code, embed, iframe	a custom tag in XmlText, or a dedicated class	decide what is safe to keep

Each datatype reads a string in its own format when you import through `fromString()`: the table is in Import and export tools, section 4.

3. Objects, versions and translations

- One source item becomes one **object**. Give it a **remote id** made from the source id (`wp-post-1234`): imports can then update instead of duplicating, and relations and redirects can be resolved later.
- History: the import creates a first version. Old revisions are rarely worth importing; archive them instead.
- **Translations** are versions of the same object in other languages. Set up the languages first (**Setup > Languages**), import the main language, then add each translation to the same object (the `language` parameter of `updateAndPublishObject()`). A source that stores each language as a separate item needs a key that links the translations (WordPress plugins, Drupal's translation sets, a shared slug) before the import.
- *Always available*: an object can be shown in languages it is not translated into; decide per class.

4. Locations: the content tree

Exponential places every object in a tree. The tree is the site's navigation, its URL structure and the scope of many permissions.

- Turn the source's main hierarchy (page tree, category tree, menu) into **folders** or landing objects.

- Give each object one **main location**; extra categories can become **additional locations** (the object appears in several places, with one canonical address) or keywords/tags.
- Sort order is per parent node (by priority, name, date ...): map the source's menu order to node priority.

5. Sections and object states

- **Sections** group content for access and design (for example "Members only", "Intranet"). Map restricted areas of the source to sections, and the roles to policies limited by section.
- **Object states** model statuses beyond draft/published (for example "In review", "Archived"). Object states explains them.

6. Users, groups, roles and passwords

- Users are objects of the `user` class in **user groups**; members and editors usually go into separate groups.
- Roles come from the source's roles or capabilities, rebuilt as Exponential roles with policies (Roles and policies). Do not copy permissions one to one; design them for the new site.
- **Passwords.** Exponential checks the hash types listed in 17.7. A source that stores **standard PHP** `password_hash()` **hashes** (they start with `$2y$`, `$argon2i$` or `$argon2id$`) can be imported as type 7 (`php_default`): the kernel checks it with `password_verify()`, and re-hashes it on the next sign-in. Hashes in any other scheme (salted MD5 or SHA variants, framework-specific formats, hashes with a prefix of their own) cannot be verified. For those, import the accounts without a usable password and send password reset mails, or keep the old login available for a transition through a custom login handler (development work).
- Personal data: import only what the new site needs, and only with a legal basis.

7. Media

- Import media as objects (image, file, media classes) below a media folder, then relate or embed them. This makes them reusable and lets editors replace one file everywhere.
- Keep the original file name and the alternative text (images whose class requires an alternative text need one).
- Images are scaled by **image aliases** (`image.ini`), not stored in every size; import the largest original.
- Rewrite links to old media addresses inside rich text to embeds of the new objects, or redirect the old media addresses.
- The upload page's ZIP importer can bring a media library over in bulk (tools).

8. Taxonomy

Source	Exponential
A category tree that is also the site's navigation	locations (folders)

Source	Exponential
Flat tags	`ezkeyword`, or `eztags`
Controlled vocabularies, hierarchical tags, synonyms	`eztags`
Categories used only for filtering	`eztags`, or object relations to category objects

9. Comments, forms, products

- **Comments**: import as child objects of a comment class of your own, or archive them. Decide whether old comments are worth moving at all.
- **Forms and submissions**: forms become classes with information collection; submissions are best archived as a file. `bccie` exports collected information as CSV.
- **Products**: classes with `ezprice`/`ezmultiprice` and the kernel shop, or an external shop linked from the site. Orders and customers usually stay in the old system's archive.

10. URLs

- A node's address comes from its name and its parent's addresses (the class's URL alias pattern can use another attribute). When the new tree mirrors the old paths, most addresses stay the same.
- Old addresses that differ get a **redirecting custom URL alias** on the node, or a **URL wildcard** for a pattern; addresses with query strings are redirected in the web server (checklist, section 4).
- Each language can have its own address of the same node.

11. Search engine metadata

- `xrowmetadata` adds a metadata attribute (title, keywords, description) and generates XML sitemaps; map the source's SEO title and description to it.
- Canonical links, `hreflang` and structured data are written by the pagelayout templates.

12. Dates and authors

- After the import, set the original publication date: the object's `published` and `modified` attributes can be set and stored by the import script; check the result in the admin.
- Map source authors to user objects (import or create them first) and set the object's owner (Changing owners), or keep the author's name in an `ezauthor` attribute.

> **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

This page is part of the Exponential documentation and is free documentation under the GNU

Free Documentation License 1.3 or any later version (the license of this documentation).

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.