Exponential Basic and Exponential 6

Moving a site between Exponential Basic 2.4 (the modernized classic 2.x CMS) and Exponential 6 - two different code bases and data models, so the content is converted by script - with the Basic tables, the article...

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

Exponential Basic (`se7enxweb/exponentialbasic`, latest release `v2.4.0.3` of 2026-08-18) modernizes the classic Exponential **2.x** CMS for PHP 8 (`php ^8.2`). It has its own kernel modules (`ezarticle`, `ezimagecatalogue`, `ezfilemanager`, `ezforum`, `eztrade`, `ezcalendar`, `ezpoll`, `ezlink`, `ezurltranslator`, `ezuser` and others) on its own tables. Its README calls one of its aims "future portability (export to another cms)".

Exponential 6 is a different code base with a different content model (classes, objects, a content tree). There is no shared database and no converter between the two in either repository; a migration is a **scripted content conversion**. This page tells you where the data is and how it maps.

Migration hub · Checklist · Import and export tools · Mapping content to Exponential

Contents

- 1. What Basic can give you
- 2. Mapping Basic to Exponential 6
- 3. From Basic to Exponential 6, step by step
- 4. From Exponential 6 to Basic
- 5. Verify, roll back, risks
- 6. Checklist

1. What Basic can give you

Read from the `main` branch of `se7enxweb/exponentialbasic`:

Source	What it is	Use
The database	tables per module, defined in `kernel/<module>/sql/<engine>/<module>.sql` (MySQL, PostgreSQL, SQLite, Informix)	the main source: read it with SQL
`bin/shell/php/ezpcsvimport.php`	a CSV utility class (any separator, quoted values, multi-line fields)	a helper if you write the export in Basic itself
RSS of articles (`kernel/ezarticle/user/articlelistrss.php`)	an RSS list of articles	only the newest items; not a full export

Source	What it is	Use
XML-RPC (`index_xmlrpc.php`, `kernel/<module>/xmlrpc/`)	article, category, image and file services	remote reading; check what each call returns before relying on it
The article "export" view (`kernel/ezarticle/admin/export.php`)	a print/typesetting output	not a data export
Storage directories (`ezimagecatalogue/catalogue/`, the file manager's files)	the image and file originals	copy them with the export

The tables that matter for most sites:

Table	Holds
`eZArticle_Article`	articles: `ID`, `Name`, `Contents` (the article markup), `AuthorID`, `Created`, `Modified`, `Published`, `IsPublished`, `Keywords`, `TopicID`, `StartDate`, `StopDate`, `ImportID`
`eZArticle_Category`, `eZArticle_ArticleCategoryLink`, `eZArticle_ArticleCategoryDefinition`	the category tree, an article's categories and its main category
`eZArticle_ArticleImageLink`, `eZArticle_ArticleFileLink`, `eZArticle_ArticleMediaLink`	an article's images, files and media
`eZArticle_Attribute`, `eZArticle_AttributeValue`, `eZArticle_Type`	article types with extra attributes
`eZImageCatalogue_Image`, `eZImageCatalogue_Category`, `eZImageCatalogue_ImageCategoryLink`	the image library
`eZUser_User`, `eZUser_Group`, `eZUser_UserGroupLink`, `eZUser_Author`	accounts, groups, authors
`eZURLTranslator_URL`	the site's own URL translations (old addresses)
`eZArticle_ArticlePermission`, `eZArticle_CategoryPermission`	read and write rights per group

The article markup. `Contents` holds an XML document written by a generator, for example the "tech" generator (`kernel/ezarticle/classes/eztechgenerator.php`): `<article><generator>tech</generator><intro>...</intro><body><page>...</page></body></article>`, with inline tags such as `<bold>`, `<italic>`, `<underline>`, `<strike>` and image references. Other generators exist (`ezsimplegenerator.php`, `ezezgenerator.php`, `ezqdomgenerator.php`); read the `<generator>` element of each article and convert per generator.

2. Mapping Basic to Exponential 6

Basic	Exponential 6
Article (`eZArticle_Article`)	an `article` class: `title` (`ezstring`) from `Name`, `intro` and `body` (`ezxmltext`) from `<intro>` and `<body>`

Basic	Exponential 6
Article pages (<page>)	one body with headings, or child objects per page when the pages are long
Article type with attributes	a class per type, one attribute per `eZArticle_Attribute`
Category tree	folders in the content tree; main category = main location, others = extra locations
Keywords	`ezkeyword` or `eztags`
Topic	`eztags`, or an object relation to topic objects
`Published`, `Created`, `Modified` (Unix timestamps)	the object's `published` and `modified`
`StartDate`, `StopDate`	publish and hide by date with workflow or cron, or plain `ezdatetime` attributes
Image catalogue image	an `image` object below a media folder; article image links become relations or embeds
File manager file	a `file` object
Forum, poll, calendar, trade	each needs its own decision: Exponential has no forum or poll in the kernel; the shop is the kernel's own; archive what you do not move
`eZUser_User`	a `user` object in user groups; `FirstName`, `LastName`, `Email`, `Login`
Group permissions	roles with policies limited by section or subtree
`eZURLTranslator_URL` and old article addresses	redirecting URL aliases or wildcards (checklist, 4)

Passwords. `kernel/ezuser/classes/ezuser.php` of Basic stores either MySQL's `PASSWORD()` format (`'*' . strtoupper( sha1( hex2bin( sha1( $password ) ) ) )`) or `md5( $password )`, depending on the database. Exponential 6 knows `md5_password` (type 1, `md5( password )`) and `mysql` (type 4, computed with the SQL function `PASSWORD()`, which MySQL 8.0 removed). So:

- MD5 hashes from Basic can be imported as type 1; Exponential re-hashes them to type 7 at the next sign-in.
- `PASSWORD()` hashes work as type 4 only while the Exponential site runs on a database server that still has `PASSWORD()` (MariaDB does; MySQL 8.0 does not). Otherwise plan password resets.

Test both on a copy before you promise users that their passwords carry over.

3. From Basic to Exponential 6, step by step

1. **Inventory** the modules in use and count rows per table; list the article generators in use (`SELECT ... FROM eZArticle_Article` and read the `<generator>` element).
2. **Install Exponential 6** and create the classes of 2 (keep their identifiers stable), the folders of the category tree, the languages, sections and roles.
3. **Export from Basic**: a script that reads the tables (SQL against a copy of the Basic database) and writes one JSON or CSV row per item, with the source id, the parent category, the converted body and the old address.

4. **Convert the article markup** to Exponential's simplified input (`<p>`, `<strong>`, `<em>`, `<a href>`, `<h1>`-`<h6>`, lists, tables): `<bold>` to `<strong>`, `<italic>` to `<em>`, image references to embeds of the image objects created in step 5.
5. **Import media first** (image catalogue and files), with remote ids such as `basic-image-<ID>`; the upload page's ZIP importer or an import script.
6. **Import the articles** with remote ids `basic-article-<ID>`: xrowextract's importer with `--map` when your export is CSV/JSON in its shape, or the script skeleton.
7. **Import users** into their groups (hash types as in 2).
8. **Addresses**: for every old address (from `eZURLTranslator_URL` and the module URLs such as `/article/view/<ID>/` and `/article/articleview/<ID>/`, `kernel/ezarticle/user/datasupplier.php`), a redirecting custom URL alias on the new node, or a wildcard.
9. **Rebuild** the search index and URL aliases; run the link checker and `exp:contentcheck`.
10. **Rehearse** twice, then cut over (checklist).

4. From Exponential 6 to Basic

Basic's model is narrower (articles, image catalogue, files, users). Export from Exponential with xrowextract (leaving Exponential): one CSV/JSON per class, with object id, main node, parent node, URL alias and every attribute; convert XmlText to Basic's article markup in a script, and insert into the Basic tables through Basic's own classes (or SQL on a copy, then checked in Basic's admin). Classes that have no Basic module stay behind or become articles. Password hashes of type 7 (`password_hash()`) are not read by Basic's login code as shown above: plan resets.

5. Verify, roll back, risks

- Counts per module and category before, per class and folder after; a sample of the 50 most read articles side by side; every image of those articles visible; every old address answers 301 to the right page.
- Rollback: Basic keeps running read-only until sign-off; switching back is a DNS or proxy change.

Risk	Counter-measure
Unknown article generators	list them first; convert per generator; test each
Images referenced inside the markup	import media first, map ids to new objects
Forums, polls, shop data	decide per module: move, replace, or archive
Passwords	test both hash formats; plan resets
Old addresses	build the redirect table from the export, test it automatically

6. Checklist

☐ Modules, tables and article generators inventoried.

☐ Classes, tree, languages, sections and roles created in Exponential 6.

☐ Export script reads a copy of the Basic database; markup converted and checked.

☐ Media, then articles, then users imported, with remote ids; relations resolved.

☐ Password strategy tested; users informed.

☐ Redirects for every old address; search and aliases rebuilt.

☐ Two rehearsals, then cut-over; Basic kept read-only until sign-off.

> **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.