content_class (migration step)

Source file: doc/eng-us/examples/migration-steps/content_class.md Last changed 2026-11-04
https://doc.exponential.earth/learninminutes/exphelp/viewer/examples/migration-steps/content_class.md

content_class

Creates, updates, ensures or deletes a content class with its attributes, their datatype settings and the translations of their names.

A step of a YAML migration with `type: content_class`. The class `expMigrationStepContentClass` (`kernel/classes/migration/steps/expmigrationstepcontentclass.php`) is registered in `settings/migration.ini` `[StepSettings] StepTypeList[content_class]`.

Modes

`mode:` is one of `create`, `update`, `ensure`, `delete`; every step names one.

Keys

Key	Type	Modes	Default	What it is
`class`	`identifier` (class)	update, delete		the class to update or delete: its identifier, id or remote:<remote id>
`identifier`	`string`	create, update, ensure		the identifier of the class; ensure finds the class by it (update: a new identifier)
`remote_id`	`string`	create, ensure		the remote id of a new class (default: a new one)
`language`	`string`	create, update, ensure		the language of name: (default: the default language of content, ContentObjectLocale)
`name`	`string`	create, update, ensure		the name of the class in the language of language:
`names`	`map`	create, update, ensure		language => the name of the class in it: its translations
`description`	`string`	create, update, ensure		the description of the class in the language of language:
`descriptions`	`map`	create, update, ensure		language => the description of the class in it
`name_pattern`	`string`	create, update, ensure		how an object is named from its attributes (<title>; default: the first attribute)

Key	Type	Modes	Default	What it is
`url_alias_pattern`	`string`	create, update, ensure		how the URL alias of an object is made (default: as its name)
`container`	`bool`	create, update, ensure		whether objects of the class can have children
`always_available`	`bool`	create, update, ensure		whether a new object of the class is shown in a language it does not have
`sort_field`	`string`: `path`, `published`, `modified`, `section`, `depth`, `class_identifier`, `class_name`, `priority`, `name`	create, update, ensure		how the children of a new object are sorted
`sort_order`	`string`: `asc`, `desc`	create, update, ensure		asc or desc
`groups`	`list`	create, update, ensure		the class groups the class is in: names or ids (create, ensure: default Content; update and ensure add the class to those it is not in)
`attributes`	`list`	create, ensure		the attributes, each a map: identifier, type (the datatype), name, names, description, descriptions, required, searchable, translatable, information_collector, category, placement, settings; ensure adds the missing ones and changes what differs
`add_attributes`	`list`	update		attributes to add, as in attributes:; every object of the class gets them
`update_attributes`	`map`	update		attribute identifier => what changes (the keys of attributes: but identifier and type)
`remove_attributes`	`list`	update		the identifiers of attributes to remove, with their data in every object
`with_objects`	`bool`	delete	`false`	true: a class with objects is deleted with them; false (the default): it is not deleted
`missing`	`string`: `fail`, `skip`	update, delete	`"fail"`	what happens when the class is not there: fail (the default) or skip

Besides these, every step takes the keys of the engine: `type`, `mode`, `if`, `references`, `environments`, `engines` and `comment` (the file format). A key of type `identifier` names a node, an object or another part of the installation by id, remote id, path or name (identifiers).

Values for references

`references: { <name>: <value> }` keeps a value of the step's result as `${ref.<name>}` for the steps and migrations after it:

Value	What it is
`class_id`	the id of the class
`identifier`	its identifier
`remote_id`	its remote id
`name`	its name
`created`	true when the step created the class
`changed`	true when the step changed it (created, updated or deleted)

Undoing

Some steps of this type undo themselves when their migration is rolled back without `down:` (the class says which); a migration with any other step of it needs `down:` or a rollback to its caption.

Caches it touches (cleared after the run): `classid`, `content`, `sortkey`.

Example

YAML

```yaml
- type: content_class
  mode: ensure
  identifier: news_article
  name: News article
  names: { ger-DE: Nachrichtenartikel }
  name_pattern: "<title>"
  groups: [ Content ]
  attributes:
    - { identifier: title, type: ezstring, name: Title, names: { ger-DE: Titel }, required: true,
 searchable: true, settings: { data_int1: 255 } }
    - { identifier: body, type: ezxmltext, name: Body }
  references: { news_class: class_id }
```

Related

- All step types
- YAML migration steps

Generated by `./console exp:help:examples` from `settings/migration.ini`, `kernel/classes/migration/steps/expmigrationstepcontentclass.php`. Do not edit this page: change the source and generate again (how). Free documentation: GFDL 1.3 or later.

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.