Fixed IDs and numeric codes

The node, object and section IDs a new installation starts with, the numbers of the sort fields and sort orders, and the status codes of objects and versions, with the setting or constant each one comes from.

Source file: `doc/eng-us/help/cms-developers/fixed-ids-and-numeric-codes.md` Last changed 2026-11-04
https://doc.exponential.earth/learninminutes/exphelp/viewer/help/cms-developers/fixed-ids-and-numeric-codes.md

This is a lookup page. Settings, fetch parameters, the database and imports use numbers where the administration shows names; the tables below translate them. Values marked *setting* can differ per site, so read them from the setting in code instead of writing the number.

The top of the tree

The tree has a root node, ID 1, which has no object and cannot be removed or shown. Below it are the top-level nodes; each starts a branch with its own section and top menu tab.

Node	Branch	Setting (`content.ini [NodeSettings]`)	Section
1	the root, no object	none	none
2	Content	`RootNode`	Standard (1)
5	Users	`UserRootNode`	Users (2)
43	Media	`MediaRootNode`	Media (3)
48	Setup	`SetupRootNode`	Setup (4)
58	Design	`DesignRootNode`	Design (5)

Top-level nodes cannot be removed or moved: the permission check refuses both for any node whose parent is the root. In templates read them with `ezini( 'NodeSettings', 'RootNode', 'content.ini' )` rather than writing 2.

Users and user groups of a new installation

These IDs come from the clean installation data. Site packages may add groups or rename them, so check your own site under **Users**.

Object ID	Node ID	What	Setting
4	5	the group Users, top of the user tree	
10	45	the anonymous user, the identity of every visitor who is not logged in	`site.ini [UserSettings] AnonymousUserID`
11	12	the group where self-registered users are created (Guest accounts in the clean data, often renamed Members)	`site.ini [UserSettings] DefaultUserPlacement` (the **node** ID)
12	13	Administrator users	
13	14	Editors	

Object ID	Node ID	What	Setting
14	15	the first administrator account	
42	44	Anonymous Users, the group of the anonymous user	

The anonymous user and the first administrator must never be removed: the site uses the first for every visitor, and the second may be the only account that can reach the setup pages.

The classes behind them are `site.ini [UserSettings] UserClassID=4` (User) and `UserGroupClassID=3` (User group).

Sections

ID	Name	Identifier
1	Standard	`standard`
2	Users	`users`
3	Media	`media`
4	Setup	`setup`
5	Design	`design`

Section IDs are never reused after a section is removed. Override rules can match either the ID (`Match[section]`) or the identifier (`Match[section_identifier]`); the identifier survives a move to another installation, the ID may not.

Sort fields and sort orders

Each node stores how its children are sorted: a field (`sort_field`) and a direction (`sort_order`). The numbers are constants of `eZContentObjectTreeNode`; `$node.sort_array` turns them into the form a fetch takes.

`sort_field`	Constant	Sorts by	Name in a fetch `sort_by`
1	`SORT_FIELD_PATH`	place in the tree	`path`
2	`SORT_FIELD_PUBLISHED`	first publication date	`published`
3	`SORT_FIELD_MODIFIED`	last modification	`modified`
4	`SORT_FIELD_SECTION`	section	`section`
5	`SORT_FIELD_DEPTH`	depth in the tree	`depth`
6	`SORT_FIELD_CLASS_IDENTIFIER`	class identifier	`class_identifier`
7	`SORT_FIELD_CLASS_NAME`	class name	`class_name`
8	`SORT_FIELD_PRIORITY`	the priority editors set by hand	`priority`
9	`SORT_FIELD_NAME`	object name	`name`

sort_field	Constant	Sorts by	Name in a fetch sort_by
10	SORT_FIELD_MODIFIED_SUBNODE	latest change anywhere below the node	modified_subnode
11	SORT_FIELD_NODE_ID	node ID	node_id
12	SORT_FIELD_CONTENTOBJECT_ID	object ID	contentobject_id
13	SORT_FIELD_VISIBILITY	hidden or visible	visibility (a node's sort_array names it is_invisible, which the fetch does not accept: pass visibility yourself)

sort_order	Constant	Meaning	In a fetch
0	SORT_ORDER_DESC	descending	false()
1	SORT_ORDER_ASC	ascending	true()

A list that follows the order editors chose:

```
{def $children = fetch( 'content', 'list', hash( 'parent_node_id', $node.node_id,
                                                 'sort_by', $node.sort_array ) )}
```

Status codes

An object (`ezcontentobject.status`, constants of `eZContentObject`):

Status	Constant	Meaning
0	STATUS_DRAFT	created, never published
1	STATUS_PUBLISHED	has a published version
2	STATUS_ARCHIVED	in the trash

A version (`ezcontentobject_version.status`, constants of `eZContentObjectVersion`):

Status	Constant	Meaning
0	STATUS_DRAFT	being edited; only its creator may edit it
1	STATUS_PUBLISHED	the version readers see; one per object
2	STATUS_PENDING	sent for publishing, held back by a workflow (approval, a date)
3	STATUS_ARCHIVED	an earlier published version, kept in the history
4	STATUS_REJECTED	refused; not set by the shipped workflows, which return a denied version to draft
5	STATUS_INTERNAL_DRAFT	a draft opened but not stored yet; old ones are removed by a cronjob
6	STATUS_REPEAT	a workflow event asked to show its form again, so the version can be edited once more
7	STATUS_QUEUED	waiting in the asynchronous publishing queue

Only published objects have nodes. A draft of a new object, a pending first version and an object in the trash are not in the tree, so a fetch of nodes never returns them.

Related pages

- Objects, nodes and locations
- Versions, drafts and publishing
- Node attributes reference
- Sections
- Glossary for CMS developers
- 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.