# Content attribute and version reference

Every attribute a template can read from a content object attribute (eZContentObjectAttribute, the items of data_map) and from a version (eZContentObjectVersion), with the PHP method behind each, values from a...

Source file: doc/eng-us/help/template-language/content-attribute-reference.md Last changed 2026-11-04
https://doc.exponential.earth/exphelp/viewer/help/template-language/content-attribute-reference.md

The values of `data_map` are `eZContentObjectAttribute` objects ( `kernel/classes/ezcontentobjectattribute.php`): one field of one version in one language. A version is an `eZContentObjectVersion` (`kernel/classes/ezcontentobjectversion.php`), reached through `$object.current` or `$object.versions`. This page lists what templates can read from both, and what `content` is for each datatype.

## Contents

1. Content object attribute: database fields
2. Content object attribute: function attributes
3. What content returns, per datatype
4. Version: database fields
5. Version: function attributes
6. Notes
7. Related pages

The tables were made like the node attributes table: `definition()` listed and every name read on a development installation, here on the `title` attribute and the current version of node 96 (object 94, an `ng_article`).

## Content object attribute: database fields

| Attribute | Database column / member | Datatype | Value on a development installation (node 96) |
|---|---|---|---|
| `id` | `ID` | integer | integer 580 |
| `contentobject_id` | `ContentObjectID` | integer | integer 94 |
| `version` | `Version` | integer | integer 1 |
| `language_code` | `LanguageCode` | string | string "eng-US" |
| `language_id` | `LanguageID` | integer | integer 2 |
| `contentclassattribute_id` | `ContentClassAttributeID` | integer | integer 334 |
| `attribute_original_id` | `AttributeOriginalID` | integer | integer 0 |
| `sort_key_int` | `SortKeyInt` | integer | integer 0 |

| Attribute | Database column / member | Datatype | Value on a development installation (node 96) |
|---|---|---|---|
| sort_key_string | SortKeyString | string | string "why should you choose strength training?" |
| data_type_string | DataTypeString | string | string "ezstring" |
| data_text | DataText | text | string "Why Should You Choose Strength Training?" |
| data_int | DataInt | integer | null |
| data_float | DataFloat | float | double 0.0 |

The fields `data_text`, `data_int` and `data_float` are the raw storage. Which one a datatype uses, and how, is the datatype's business: `ezstring` keeps its text in `data_text` and leaves `data_int` empty (`null` above); `eZContentObjectAttribute::attribute()` turns a `null` `data_text` into `''` so templates never see null there.

## Content object attribute: function attributes

| Function attribute | PHP method | Value on a development installation (node 96) |
|---|---|---|
| contentclass_attribute | eZContentObjectAttribute::contentClassAttribute() | object eZContentClassAttribute |
| contentclass_attribute_identifier | eZContentObjectAttribute::contentClassAttributeIdentifier() | string "title" |
| contentclass_attribute_name | eZContentObjectAttribute::contentClassAttributeName() | string "Title" |
| can_translate | eZContentObjectAttribute::contentClassAttributeCanTranslate() | integer 1 |
| is_information_collector | eZContentObjectAttribute::contentClassAttributeIsInformationCollector() | integer 0 |
| is_required | eZContentObjectAttribute::contentClassAttributeIsRequired() | integer 1 |
| content | eZContentObjectAttribute::content() | string "Why Should You Choose Strength Training?" |
| has_http_value | eZContentObjectAttribute::hasHTTPValue() | boolean false |
| value | eZContentObjectAttribute::value() | string "Why Should You Choose Strength Training?" |
| has_content | eZContentObjectAttribute::hasContent() | boolean true |
| class_content | eZContentObjectAttribute::classContent() | string "" |
| object | eZContentObjectAttribute::object() | object eZContentObject |
| object_version | eZContentObjectAttribute::objectVersion() | object eZContentObjectVersion |

| Function attribute | PHP method | Value on a development installation (node 96) |
|---|---|---|
| view_template | eZContentObjectAttribute::viewTemplateName() | string "ezstring" |
| edit_template | eZContentObjectAttribute::editTemplateName() | string "ezstring" |
| result_template | eZContentObjectAttribute::resultTemplate() | string "ezstring" |
| has_validation_error | eZContentObjectAttribute::hasValidationError() | boolean false |
| validation_error | eZContentObjectAttribute::validationError() | null |
| validation_log | eZContentObjectAttribute::validationLog() | null |
| language | eZContentObjectAttribute::language() | object eZContentObjectAttribute |
| is_a | eZContentObjectAttribute::isA() | string "ezstring" |
| display_info | eZContentObjectAttribute::displayInfo() | array(4) of array |
| class_display_info | eZContentObjectAttribute::classDisplayInfo() | array(2) of array |

A sample:

**Template**

```
{def $attribute = fetch( 'content', 'node', hash( 'node_id', 96 ) ).data_map.title}
id / version / language: {$attribute.id} / {$attribute.version} / {$attribute.language_code}
class attribute:         {$attribute.contentclass_attribute_identifier} "{$attribute.contentclass
_attribute_name|wash}" ({$attribute.contentclassattribute_id})
datatype:                {$attribute.data_type_string}, is_a: {$attribute.is_a}
required / translatable: {$attribute.is_required} / {$attribute.can_translate}
templates:               view {$attribute.view_template}, edit {$attribute.edit_template}
stored:                  data_text "{$attribute.data_text|wash}", sort_key_string "{$attribute.so
rt_key_string|wash}"
```

Output:

```
id / version / language: 580 / 1 / eng-US
class attribute:         title "Title" (334)
datatype:                ezstring, is_a: ezstring
required / translatable: 1 / 1
templates:               view ezstring, edit ezstring
stored:                  data_text "Why Should You Choose Strength Training?", sort_key_string "w
hy should you choose strength training?"
```

## What content returns, per datatype

`content` calls `eZContentObjectAttribute::content()`, which asks the datatype's `objectAttributeContent()` once and keeps the result on the attribute. This template walked the first 150 nodes below the Fit & Healthy front page on a development installation and printed, for each datatype the first time it had content, what `content` returned:

**Template**

```
{def $seen = array()}
{foreach fetch( 'content', 'tree', hash( 'parent_node_id', 89, 'limit', 150 ) ) as $node}{foreach
 $node.data_map as $attribute}{if and( $attribute.has_content, $seen|contains( $attribute.data_ty
pe_string )|not )}{set $seen = $seen|append( $attribute.data_type_string )}{$attribute.data_type_
string}: {if is_object( $attribute.content )}object {$attribute.content|get_class}{else}{$attribu
te.content|get_type|explode( '[' )|extract_left( 1 )|implode( '' )}{/if} ({$node.class_identifier
}.{$attribute.contentclass_attribute_identifier})

{/if}{/foreach}{/foreach}
```

Output:

```
ezstring: string (ng_article.title)
ezxmltext: object ezxmltext (ng_article.teaser_intro)
xrowmetadata: object xrowmetadata (ng_article.metadata)
eztags: object eztags (ng_landing_page.tags)
ezimage: object ezimagealiashandler (ng_article.image)
ezobjectrelationlist: array (ng_article.authors)
ezdatetime: object ezdatetime (ng_blog_post.publish_date)
ezinteger: integer (ng_gallery.column_count)
ezboolean: integer (ng_category.show_children)
ezemail: string (ng_contact_form.recipient)
ngenhancedlink: object ngenhancedlinkvalue (ng_component_hero.link_1)
ezbinaryfile: object ezbinaryfile (ng_video.video_file)
ezmedia: object ezmedia (ng_audio.file)
eztext: string (ng_htmlbox.html_code)
ezurl: string (ng_site_info.facebook)
ezprice: object ezprice (product.price)
ezsrrating: object ezsrratingobject (product.rating)
ezmatrix: object ezmatrix (product.specifications)
```

| **Datatype** | `content` **on a development installation** | **Datatype page** |
|---|---|---|
| `ezstring`, `eztext`, `ezemail`, `ezurl` | string (for `ezurl` the URL) | ezstring, eztext, ezemail, ezurl |
| `ezinteger`, `ezboolean` | integer | ezinteger, ezboolean |
| `ezxmltext` | `eZXMLText` object; `.output.output_text` is the rendered HTML | ezxmltext |
| `ezimage` | `eZImageAliasHandler`; `.original`, `[alias name]` | ezimage |
| `ezobjectrelationlist` | array with the key `relation_list` | ezobjectrelationlist |
| `ezdatetime` | `eZDateTime` object | ezdatetime |
| `ezbinaryfile` | `eZBinaryFile` object | ezbinaryfile |
| `ezmedia` | `eZMedia` object | ezmedia |
| `ezprice` | `eZPrice` object | ezprice |
| `ezmatrix` | `eZMatrix` object | ezmatrix |
| `eztags` | `eZTags` object (extension eztags) | |
| `xrowmetadata` | `xrowMetaData` object (extension xrowmetadata) | |
| `ngenhancedlink` | `ngEnhancedLinkValue` object (extension exp_enhanced_link) | |

| Datatype | `content` on a development installation | Datatype page |
|---|---|---|
| `ezsrrating` | `ezsrRatingObject` (extension ezstarrating) | |

`get_class` lower-cases class names, so the example shows `ezimagealiashandler` for `eZImageAliasHandler`. The Data map and content page reads several of these contents in detail.

## Version: database fields

| Attribute | Database column / member | Datatype | Value on a development installation (node 96) |
|---|---|---|---|
| `id` | `ID` | integer | integer 544 |
| `contentobject_id` | `ContentObjectID` | integer | integer 94 |
| `creator_id` | `CreatorID` | integer | integer 14 |
| `version` | `Version` | integer | integer 1 |
| `status` | `Status` | integer | integer 1 |
| `created` | `Created` | integer | integer 1600936225 |
| `modified` | `Modified` | integer | integer 1600936234 |
| `workflow_event_pos` | `WorkflowEventPos` | integer | integer 0 |
| `user_id` | `UserID` | integer | integer 0 |
| `language_mask` | `LanguageMask` | integer | integer 2 |
| `initial_language_id` | `InitialLanguageID` | integer | integer 2 |

`status` is the version status: `eZContentObjectVersion::STATUS_DRAFT` (0), `STATUS_PUBLISHED` (1), `STATUS_PENDING` (2), `STATUS_ARCHIVED` (3), `STATUS_REJECTED` (4), `STATUS_INTERNAL_DRAFT` (5), `STATUS_REPEAT` (6), `STATUS_QUEUED` (7).

## Version: function attributes

| Function attribute | PHP method | Value on a development installation (node 96) |
|---|---|---|
| `creator` | `eZContentObjectVersion::creator()` | object eZContentObject |
| `name` | `eZContentObjectVersion::name()` | string "Why Should You Choose Strength Training?" |
| `version_name` | `eZContentObjectVersion::versionName()` | string "Why Should You Choose Strength Training?" |
| `main_parent_node_id` | `eZContentObjectVersion::mainParentNodeID()` | integer 92 |
| `contentobject_attributes` | `eZContentObjectVersion::contentObjectAttributes()` | array(18) of eZContentObjectAttribute |
| `related_contentobject_array` | `eZContentObjectVersion::relatedContentObjectArray()` | array(0) |

| Function attribute | PHP method | Value on a development installation (node 96) |
|---|---|---|
| reverse_related_object_list | eZContentObjectVersion::reverseRelatedObjectList() | array(0) |
| parent_nodes | eZContentObjectVersion::parentNodes() | array(1) of eZNodeAssignment |
| can_read | eZContentObjectVersion::canVersionRead() | boolean false |
| can_remove | eZContentObjectVersion::canVersionRemove() | boolean false |
| data_map | eZContentObjectVersion::dataMap() | array(18) of eZContentObjectAttribute |
| node_assignments | eZContentObjectVersion::nodeAssignments() | array(1) of eZNodeAssignment |
| contentobject | eZContentObjectVersion::contentObject() | object eZContentObject |
| initial_language | eZContentObjectVersion::initialLanguage() | object eZContentLanguage |
| language_list | eZContentObjectVersion::translations() | array(1) of eZContentObjectTranslation |
| translation | eZContentObjectVersion::translation() | boolean false |
| translation_list | eZContentObjectVersion::defaultTranslationList() | array(1) of eZContentObjectTranslation |
| complete_translation_list | eZContentObjectVersion::translationList() | array(1) of eZContentObjectTranslation |
| temp_main_node | eZContentObjectVersion::tempMainNode() | object eZContentObjectTreeNode |

A sample:

**Template**

```
{def $version = fetch( 'content', 'node', hash( 'node_id', 96 ) ).object.current}
version {$version.version}, status {$version.status}, created {$version.created|datetime( 'custom
', '%Y-%m-%d' )}
name in this version: {$version.version_name|wash}
translations: {foreach $version.translation_list as $t}{$t.language_code}{delimiter}, {/delimiter
}{/foreach}

creator object id: {$version.creator_id}
```

Output:

```
version 1, status 1, created 2020-09-24
name in this version: Why Should You Choose Strength Training?
translations: eng-US
creator object id: 14
```

## Notes

- `language` returns the same attribute in another language: without a language code, the object's initial language (`eZContentObjectAttribute::language()`).
- `view_template` and `edit_template` are the names (without `.tpl`) that `attribute_view_gui` and `attribute_edit_gui` look for below `content/datatype/view/` and `content/datatype/edit/`; for an information collector attribute `view_template` returns the collector template instead.
- `class_content` is the content of the class attribute (the field's settings, such as a selection's options).
- `has_content` comes from the datatype's `hasObjectAttributeContent()`. Most datatypes return a boolean; the `ezimage` datatype returned an empty string for an image field without an image on a development installation (see The template compiler for why that matters with `choose`).
- The `can_read` and `can_remove` of a version check the current user's `content/versionread` and `content/versionremove` policies; as the anonymous user both were false.

## Related pages

- The object model in templates
- Data map and content
- Node attributes reference
- Content object attributes reference
- Datatype view templates
- Datatypes with examples
- 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.