
eZ OpenDocument Format (eZODF)
Extension

Documentation

version 2.0

Table of Contents

1.Introduction.....	3
1.1.Target audience.....	3
1.2.Conventions.....	3
1.3.More resources.....	3
1.4.Contacting eZ.....	4
1.5.Copyright and trademarks.....	4
2.Requirements.....	4
3.Installation.....	4
3.1.Microsoft Word import configuration.....	4
3.1.1.Installation of OpenOffice.org macros.....	5
3.1.2.Starting the document conversion daemon.....	5
4.Importing.....	5
4.1.Configuring imported documents.....	7
4.2.Supported datatypes.....	7
4.3.Supported formatting.....	7
5.Exporting.....	8
5.1.Supported datatypes.....	8
5.2.Supported formatting.....	8
5.3.Export action.....	8
5.4.Export templates.....	10



1. Introduction

The eZ OpenDocument Format (eZODF) extension is an integration between eZ Publish (versions 3.8 and higher) and OpenDocument Text (.odt) documents. It supports both importing and exporting .odt documents.

1.1. Target audience

This manual describes how to perform basic importing and exporting operations with the eZODF version 2.0 extension. It is intended for anyone working with this extension in the Administration Interface. For information on how to use and navigate the Administration Interface, refer to the eZ Publish User Manual at <http://ez.no/doc>.

1.2. Conventions

- Code samples, functions, variable names, and on are printed in `monospace font`.
- Filenames and paths are printed in *monospace italic font*.
- Commands are printed in **monospace bold font**.
- Elements of graphical user interfaces (such as buttons and field labels) are printed in **bold font**.
- Component names (such as an application) are capitalized, for example “Administration Interface”.
- In sample URLs, replace “example.com” with the domain name of your site.
- The screenshots in this document might have been modified to fit the page or to illustrate a point, and therefore might not exactly match the display on your site.

1.3. More resources

For assistance with eZODF or eZ Publish, refer to the following resources:

- **eZ Publish documentation:** eZODF is an extension to eZ Publish. Documentation for shared features is not reproduced in these manuals. Instead, where appropriate, there are links in this document to the online versions of the eZ Publish documentation, located at <http://www.ez.no/doc>.
- **eZ Publish forums:** The forums on the eZ Systems website are a valuable community-driven resource, where eZ Publish users provide assistance and support to each other. Accessing the forums is free. The forums are located at <http://ez.no/community/forum>.
- **Support from eZ Partners:** eZ's global network of partners provide professional assistance for all eZ products. To find a partner, contact sales@ez.no.
- **Other eZ solutions:** For information about other solutions provided by eZ Systems, refer to <http://ez.no/products/solutions>.
- **Training and certification:** eZ Systems and eZ Partners offer training courses and certifications for eZ Publish. Contact sales@ez.no or visit <http://ez.no/services/training> for more information.



1.4. Contacting eZ

For non-technical questions regarding eZODF, eZ Publish or eZ Systems, please contact us:

- <http://ez.no/company/contact>
- info@ez.no

We welcome your suggestions and comments on this manual. Please write to team-knowprod@ez.no.

1.5. Copyright and trademarks

Copyright © 2007 eZ Systems AS. Permission is granted to copy, distribute and/or modify this document under the terms of the [GNU Free Documentation License](#), Version 1.2 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](#)".

Other product and company names mentioned in this manual may be the trademarks of their respective owners. We use trademark names in an editorial fashion to the benefit of the trademark holder; therefore, these names are not marked with trademark symbols. All terms known to be trademarks have been appropriately capitalized. We cannot attest to the accuracy of this usage, and usage of a term in this book should not be regarded as affecting the validity of any trademark or servicemark.

2. Requirements

This extension supports eZ Publish 3.8 and later. It might work on earlier versions back to 3.5, but this is not officially supported or recommended.

Either the zlib extension must be enabled in your PHP installation or you need to have the command line versions of zip and unzip available on your operating system. This is needed to create and unpack the zip files that make up OpenDocument Text (.odt) documents.

3. Installation

Unpack the tar.gz archive and place the *ezodf* folder under the *extension* folder in eZ Publish. Enable the eZODF extension in the eZ Publish Administration Interface under the **Setup->Extensions** menu.

Note: if you want to upload large documents you need to increase the upload memory limit in PHP. This is configured in the *php.ini* configuration file.

3.1. Microsoft Word import configuration

To install Microsoft Word import support you need to do the following on your web server:

1. Install OpenOffice.org 2.x
2. Install document conversion macros in OOo
3. Start the document conversion daemon

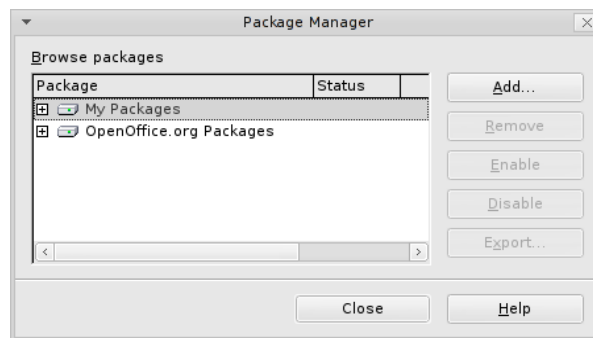
The reason you need to have OpenOffice.org installed is that eZ Publish uses it to first



convert the Word file formats to OpenDocument Text format, which is then imported. Tested formats include .doc and .rtf.

3.1.1. Installation of OpenOffice.org macros

Go to the package manager interface in OpenOffice.org. This is found under **Tools -> Package Manager...**



Choose **My Packages** and click **Add...**

Then, select the file `extension/ezodf/doc/ezconversion.zip`.

3.1.2. Starting the document conversion daemon

The document conversion daemon is a PHP script bundled with the extension. It communicates with OpenOffice.org and does the actual conversion of documents.

The script needs to be started with the CLI version of PHP, which means that you need to execute the following command from the command line:

```
$ php extension/ezodf/daemon.php
```

Note: You must have compiled PHP with the following switches:

```
--enable-cli  
--enable-pcntl  
--enable-sockets
```

Also, the document conversion daemon will only run under Linux, as socket support in PHP does not work under Windows.

4. Importing

To import OpenDocument Text (.odt) documents to eZ Publish you need to access the URL `example.com/index.php/ezodf/import`. Replace the `example.com/index.php` part with the correct URL for the **Administration Interface** of your eZ Publish site.

The import screen is shown below. Here you browse for the .odt file to import, then click the **Upload file** button.



OpenOffice.org import

Import OpenOffice.org document

You can import OpenOffice.org Writer documents directly into eZ publish from this page. You are asked where to place the document and eZ publish does the rest. The document is converted into the appropriate class during the import, you get a notice about this after the import is done. Images are placed in the media library so you can re-use them in other articles.

After uploading the file, you need to place it in the eZ Publish content node tree. The standard eZ Publish browse window is shown below. Find the container under which you wish to import the file, select its corresponding radio button, then click the **OK** button. If you want to cancel the import, simply press the **Cancel** button.

Choose document placement

Please choose the placement for the OpenOffice.org object.

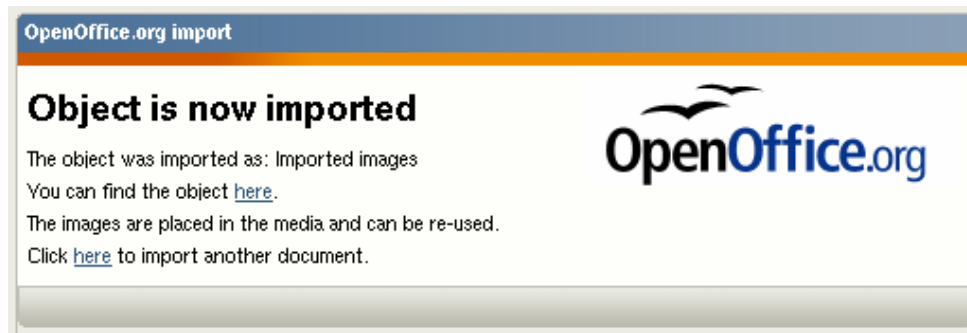
Select the placements and click the Select button.
Using the recent and bookmark items for quick placement is also possible.
Click on placement names to change the browse listing.

↑ eZ publish [8]

10 25 50

	Name	Type
<input type="radio"/>	OOo	Folder
<input type="radio"/>	Contact us	Feedback form
<input type="radio"/>	Links	Folder
<input type="radio"/>	Files	Folder
<input type="radio"/>	Galleries	Folder
<input type="radio"/>	Forums	Folder
<input type="radio"/>	Products	Folder
<input type="radio"/>	News	Folder

If the import is successful you will see the screen below. Here, there is a link to the imported object and a link to import more files.



4.1. Configuring imported documents

A standard OpenDocument Text (.odt) document imports into the default eZ Publish content class in eZ Publish. Here, the title of the object is set from the filename and the actual contents of the document are stored in the attributes defined in the *odf.ini* configuration file in the *extensions/odf/settings/* folder.

See the settings below for a configuration example where documents are imported as articles, the title is stored in the *title* attribute and the document contents are stored in the *body* attribute.

```
[ODFImport]
DefaultImportClass=article
DefaultImportTitleAttribute=title
DefaultImportBodyAttribute=body
```

You can define sections in .odt documents to map to eZ Publish attributes. For example, you can define sections in an .odt document for the title, introduction and body to be automatically imported to an article, while another document can be imported as a product.

This is configured in *odf.ini*. There are sample documents under the *examples/* folder in the eZODF extension that can be used for testing.

4.2. Supported datatypes

When importing OpenDocument Text (.odt) documents, there are some limitations on the datatypes to which it can import content. Currently the following datatypes are supported:

- Text line
- Text fields
- XML text
- Image
- Date
- Date and time
- Matrix



4.3. Supported formatting

When importing text into XML text fields, the following formatting styles are supported:

- Normal text
- Bold and italic text spans
- Text links
- Custom tags (such as a factbox)
- Numbered and un-numbered lists (one level)
- Headers (multiple levels)
- Tables (without collapsed rows or columns)
- Embedded images
 - Alignment and size are supported
 - Image captions are not supported

5. Exporting

Any content object in eZ Publish can be exported to OpenDocument Text (.odt) documents. There are however some limitations on the supported datatypes.

5.1. Supported datatypes

The following datatypes are currently supported:

- Text line
- Text field
- XML text
- Image
- Date
- Date and time
- Matrix

5.2. Supported formatting

When exporting XML text, the following formatting styles are supported:

- Normal text
- Text links
- Custom tags (such as a factbox)
- Headers
- Numbered and un-numbered lists
- Tables (without collapsed rows or columns)
- Embedded images

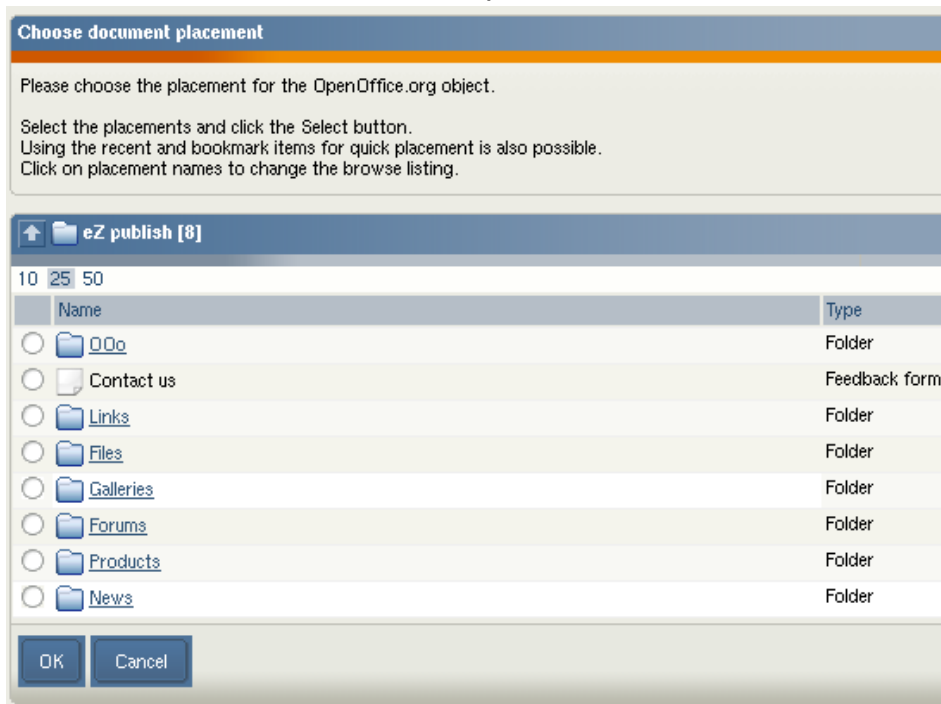


5.3. Export action

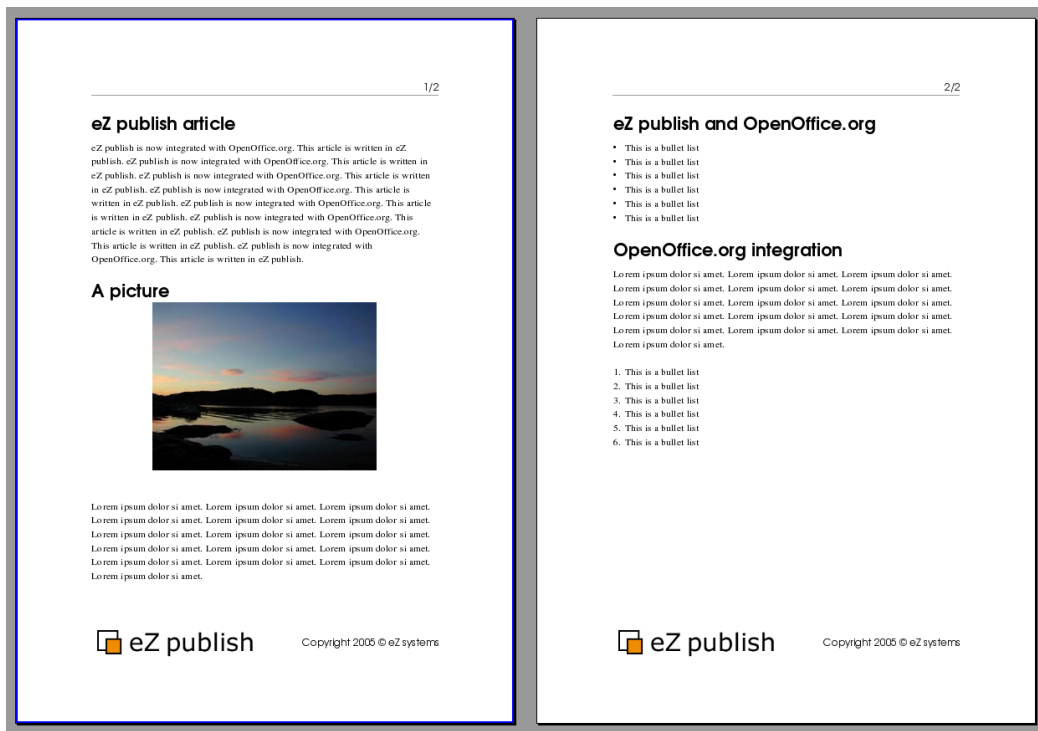
To do the actual object export you need to access the URL `example.com/index.php/ezodf/export`. Replace the `example.com/index.php` part with the correct URL for the Administration Interface of your eZ Publish site. The initial export screen is shown in the screenshot below.



Click the **Export Object** button to proceed to the browse screen shown below. Here you navigate and select the object to export. Currently you can only export one object at a time; sub items are not included in the export.



After selecting the object for export and clicking the **OK** button, eZ Publish will perform the actual export action. The document is served via the browser; you can choose to either download the document or open it directly in OpenOffice.org Writer (if it is properly installed on your computer) or a similar word processor that supports .odt documents. The screenshot below shows a preview of an exported object. Notice the headers and footers; these are defined in standard OpenOffice.org Writer templates.



5.4. Export templates

OpenDocument Text (.odt) documents can be exported using either the default OpenOffice.org styles or with a custom OpenOffice.org Writer template file. This is configured in the `ODFExport` section of the `odf.ini` configuration file. Set `UseTemplate` to `true` to enable the use of export templates. The actual template used needs to be placed under the `extension/ezodf/templates/` folder. The name of the template is specified with the `TemplateName` setting. Below is an example of this configuration.

```
[ODFExport]
UseTemplate=true
TemplateName=eztemplate.stw
```