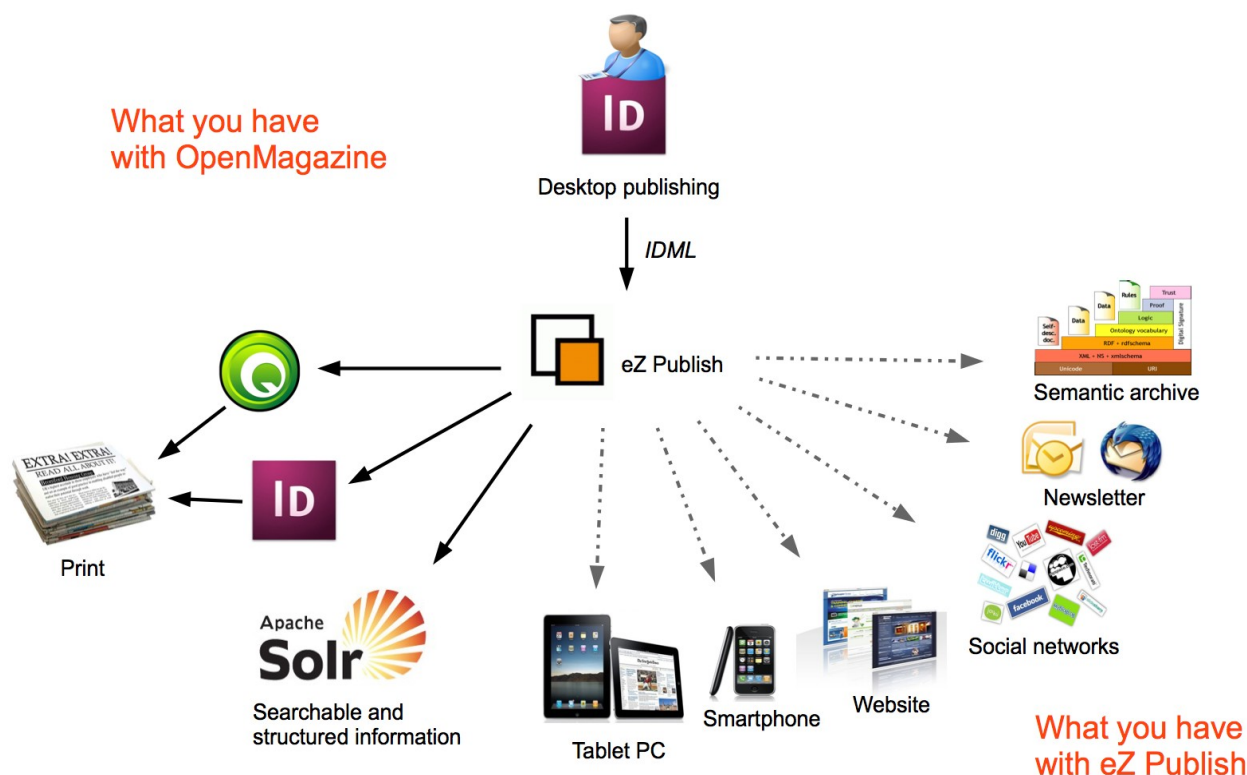




Enterprise content management
Multicanalità
Flussi di lavoro su web

OpenMagazine

user's manual



Versione 1.1, 28/08/2011

Opencontent SCARL

Via Grazioli 106, 38122 Trento, Italy – P.IVA: 02190640223

Phone: +39 04611862944

Fax: +39 04611862844

E-mail: info@opencontent.it

Web: www.opencontent.it

Summary

Introduction.....	2
How it works.....	3
Using OpenMagazine.....	4
Initial configuration of the magazine.....	4
Tagging of the boxes in the graphic structure.....	4
Magazine creation.....	5
Uploading the sections (file IDML) in eZ Publish.....	5
Laying out the magazine and validation of the information.....	8
IDML file creation with contents from eZ publish.....	9
Using OpenMagazine in an editorial context.....	9
eZTransfer.....	10
Configuration of the system.....	11
Requirements.....	11
Configuration files: openmagazine.ini.....	11
Configuration files: idml.ini.....	14
Licence.....	14
About us.....	14
Contacts.....	15

Introduction

Do you produce a magazine or daily paper and need to store your contents?

OpenMagazine was created by OpenContent and is the extension of eZ Publish. It permits the automation of a great deal of editorial work, enabling eZ Publish to utilise the formats used by the main software systems for desktop publishing.

Starting from a magazine (generated with Adobe InDesign, for example), the contents can be loaded and filed in eZ Publish. This permits a very flexible multichannel management in which contents are distributed via the web, social networks, mobile phones, newsletters, paper editions of the magazine.

Added value

Thanks to its services oriented architecture (SaaS), the OpenMagazine extension can be used both by big publishing groups and by small weeklies. The immediate benefit is the creation of a structured archive with all its informative contents accessible on-line, searchable via the web, reusable, lasting over time.

Full compatibility with Adobe InDesign

The OpenMagazine extension manages the import and export of documents in IDML format. IDML is a format created by Adobe for the InDesign suite and available starting from the CS4 version. Based on XML, IDML fully describes an InDesign document.

Using text, images and graphic elements based on XML, IDML permits the creation and modification of documents using InDesign or any other automatic system able to interpret the XML format.

Just as its forerunner INX, IDML is a DOM (Document Object Model) representation of InDesign documents and is based on the scripting object model. For further information consult the official manual:

<http://www.images.adobe.com/www.adobe.com/content/dam/Adobe/en/devnet/indesign/cs55-docs/IDML/idml-specification.pdf>

Opencontent SCARL

Via Grazioli 106, 38122 Trento, Italy – P.IVA: 02190640223

Phone: +39 04611862944

Fax: +39 04611862844

E-mail: info@opencontent.it

Web: www.opencontent.it

Features

The OpenMagazine extension permits:

- the recovery of the graphic styles and content structure designed with Adobe InDesign, in your eZ Publish installation. Simply load one or more IDML files in eZ Publish, using the interfaces commonly used in this CMS (e.g. the ezmultiupload extension)
- the association of any type of content within eZ Publish to the structures previously created with InDesign. This operation is carried out with an intuitive web interface
- the creation of IDML files that use the graphics created with InDesign and the eZ Publish contents. These files can be modified further by InDesign and thus uploaded again onto the eZ Publish platform, through the OpenMagazine extension
- the automation of the content download operation. With OpenMagazine it is possible to download the files created by the extension directly into the user's filesystem without any manual intervention through eZ Publish. This feature speeds up the *page layout* and the *data filing* processes significantly, thus OpenMagazine perfectly meets the needs of daily papers' editorial offices as well.

How it works

The standard procedure for managing a magazine or a daily paper provides for the creation of a *page planning*, containing a certain number of *articles* organised by *sections*.

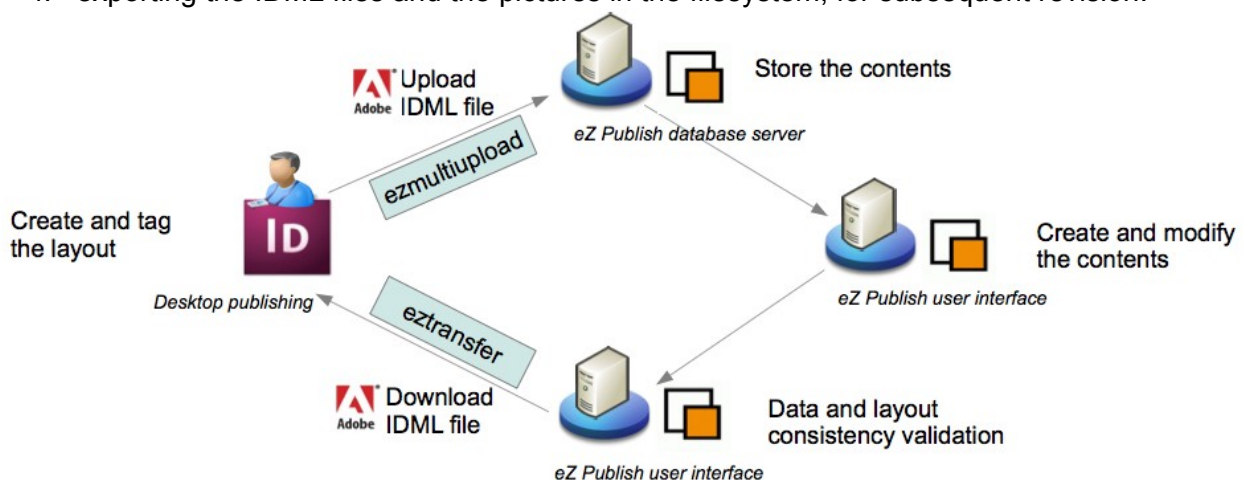
OpenMagazine reflects this approach, providing a structure based on three hierarchical levels:

- a single container ("magazine container")
- the sections of the magazine/newspaper, each one possibly characterised by its own graphics
- the contents (articles, advertising, ...)

The creation of a new magazine with OpenMagazine starts with an object of the "magazine container" type, in which objects of the "magazine section" type will be created, each one characterised by its own graphic layout. The contents to be laid out can be put in the "magazine section", or in a separate eZ Publish node recipient.

Generally editorial work will be organised according to four phases:

1. IDML file tagging, using InDesign
2. IDML file upload in eZ Publish
3. linking the eZ Publish contents to the structure of the sections and validation of the consistency of the contents themselves with the structure by using OpenMagazine
4. exporting the IDML files and the pictures in the filesystem, for subsequent revision.



Opencontent SCARL

Via Grazioli 106, 38122 Trento, Italy – P.IVA: 02190640223

Phone: +39 04611862944

Fax: +39 04611862844

E-mail: info@opencontent.it

Web: www.opencontent.it

There are, however, other possible ways of using the OpenMagazine extension. During the design phase attention was in fact paid to the main variables that affect the organisational models of media companies. In particular attention was given to the following factors:

- the organisation of the editorial office and the need to respect the different roles (director, editor in chief, heads of services, editors, etc.)
- the frequency with which publications (monthlies, weeklies, dailies) are released and distributed
- the size of the company and the role distribution
- the availability of personnel working exclusively on magazine or daily layout
- the possibility of using predefined graphic grids.

Using OpenMagazine

Following are the main steps for using the OpenMagazine extension.

Initial configuration of the magazine

The initial configuration of the magazine involves a series of operations mainly to be carried out by the graphic designer. These tasks are aimed at streamlining the magazine upload (files in IDML format) onto the eZ Publish platform.

Once one or more – potentially infinite – graphic models have been defined for the individual sections of the magazine, they are suitably tagged in order for the single elements of the section to be recognisable to eZ Publish.

In this way it is possible to create a “book” of predefined graphic models, infinitely reusable. Thus, the production process is speeded up.

Tagging of the boxes in the graphic structure

OpenMagazine is able to manage incoming information correctly, only if the contents of the initial InDesign document were relevantly tagged and exported into the filesystem in a file in IDML format.

An InDesign document can be made up by one or more stories (e.g. articles), each one of which will be characterised by a number of elements. Generally a magazine or daily article includes an subhead, a title, the body text, possibly a dossier or a summary, one or more pictures, the author, the publication date. All these elements must be relevantly tagged. The operation is very simple for those who have a basic knowledge of Adobe InDesign.

For information on the procedure for tagging the elements of an InDesign document look up the official manual:

http://help.adobe.com/en_US/InDesign/6.0/WSa285fff53dea4f8617383751001ea8cb3f-6cf5a.html

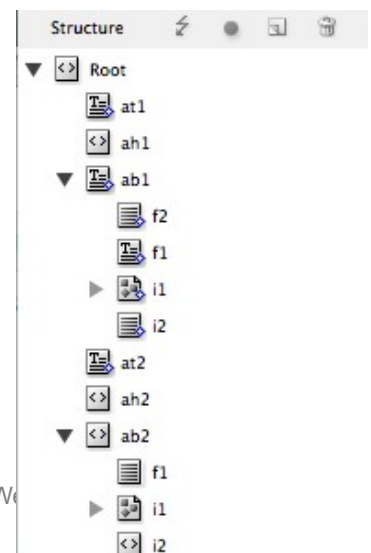
In the definition of the tags, the OpenMagazine extension uses a **convention** that in the import phase permits the linking of each tagged box to its respective attributes in eZ Publish.

The convention used by OpenMagazine for interpreting the tags is based on two criteria:

- Tag names
- Tag hierarchy

By convention, the names of the tags are made up of 3 elements:

- identifying letter of the eZ Publish class
- identifying letter of the attribute of this class
- identifying number of the story, that must be unambiguous within the story itself. This number will be used by OpenMagazine to:
 - group the attributes together



Opencontent SCARL

Via Grazioli 106, 38122 Trento, Italy – P.IVA: 02190640223

Phone: +39 04611862944

Fax: +39 04611862844

E-mail: info@opencontent.it

W

- organise the priorities of the eZ Publish contents in the document; this number in fact corresponds to the priority of the eZ Publish object within the container in which it is located.

The tag hierarchy is very simple:

- all the attributes of the story must be placed on the first level (directly under the root of the tree structure)
- the pictures and the factboxes must be put under in tags containing formatted texts (that in eZ Publish will be linked to the datatype ezxmltext), as children.

In particular:

- the pictures must be tagged simply by using the letter "i", followed by a number that identifies them unambiguously in the text box and determines the order in which they will be displayed;
- the factboxes must be tagged with the letter "f", followed by a number that identifies them unambiguously in the text box and determines the order in which they will be displayed;

For example, if an InDesign document were made up of 2 stories, each one characterised by title, sub head, body text, factboxes and pictures, the tag names could be:

- at1 (title of the first article)
- ah1 (sub head of the first article)
- ab1 (text body of the first article)
 - i1 (first picture of the first article), placed within the ab1tag
 - i2 (second picture of the first article), placed within the ab1tag
 - f1 (first factbox of the first article), placed within the ab1tag
- f2 (second factbox of the first article), placed within the ab1tag
- at2 (title of the second article)
- ah2 (sub head of the second article)
- ab2 (body text of the second article)
 - i1 (first picture of the second article), placed within the ab2 tag
 - i2 (second picture the second article), placed within the ab2 tag
 - f1 (first factbox of the second article), placed within the ab2 tag

All the letters used in the tags can be personalised by the user within the configuration file (openmagazine.ini); in this way it is possible to choose the letters that are most significant for the user (e.g.: class and attribute initial, in one's own mother tongue), and thus easier to remember. This group of factors and criteria will enable an unambiguous definition of the contents of each tagged box and take it directly into eZ Publish.

Magazine creation

Uploading the sections (file IDML) in eZ Publish

There are three possible scenarios:

- the IDML file created by Adobe InDesign only contains the graphic structure and the page styles, with some place holder text: in this case only the structure, the styles and the bulk of the sections will be imported into eZ Publish; the editor will later decide on the source of information to be linked to these sections, by selecting a node recipient (e.g. folder) from eZ Publish's content tree;
- the IDML file contains both the graphic styles and the contents (texts and references to the pictures); in this case the structure, the styles, the bulk and the contents will be imported in a single operation, creating new objects in eZ Publish. The object classes that can be imported into eZ Publish (article, blog_post, image, ...) can be defined by the user in the

Opencontent SCARL

Via Grazioli 106, 38122 Trento, Italy – P.IVA: 02190640223

Phone: +39 04611862944

Fax: +39 04611862844

E-mail: info@opencontent.it


Web: www.opencontent.it

- specific configuration file;
- the IDML file was originally produced by eZ Publish and was subsequently modified with Adobe InDesign: it is probable that the contents of the new file are different from those in the eZ Publish objects. In this case, when the IDML file modified with InDesign is uploaded, a new version of the objects already present in eZ Publish will be created.

Loading one or more IDML files onto eZ Publish is very easy:

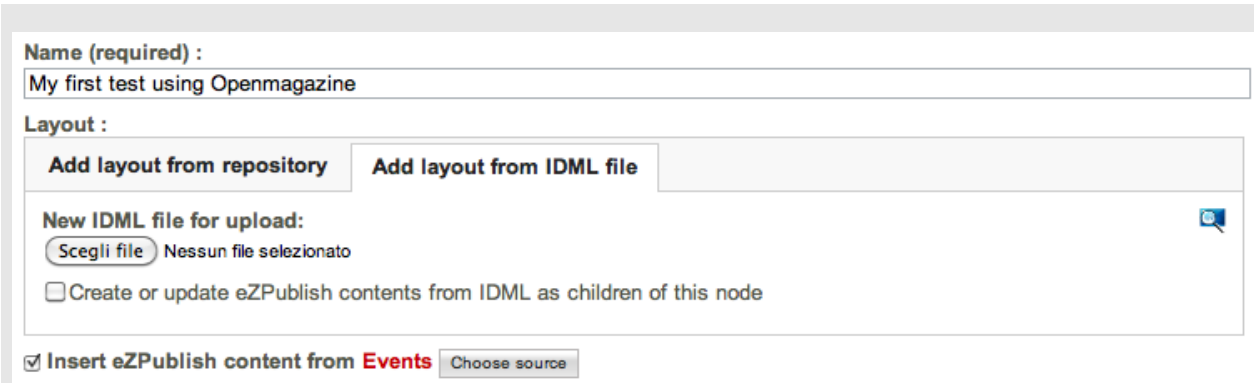
- first of all an object container is defined (e.g. a “magazine container” or a simple folder) in which the various sections of the magazine are to be put; each section (magazine section) will be characterised by its own graphics (imported by the IDML file);
- to **load a single IDML file**, a new object of the “magazine section” class must be created in eZ Publish, specifying its title and selecting the IDML file in the filesystem;
- to **load more than one IDML file at the same time**, use the eZ Publish *ezmultiupload* standard function, which will automatically create “magazine section” type objects.

In this way a graphic layout created by InDesign is associated with a “magazine section” type object.

In the upload phase, it's also possible to see a preview of the section, by pressing the icon “Preview”: 

Checking the box “**Create or update eZ Publish contents from IDML as children of this node**”, not only the styles, but also the contents of the InDesign document (texts and pictures) are imported into eZ Publish, updating the objects possibly already in the database.

Selecting a node with the button “**Insert eZ Publish content from [Choose source]**” it's possible to personalise the source of the contents to be inserted into the section; if the source isn't specified, the contents of the section itself will be considered.



With a single operation, the interface for the editor permits the manual upload of the sections (the IDML files), the preview and the selection of the source node

It's also possible to archive a certain number of graphic layouts within an eZ Publish archive, in order to be able to reuse them when needed during the composition of new editions of the magazine. With the same upload procedure, the graphic layouts (IDML) can be put in the specific repository (a node container defined in the configuration file openmagazine.ini).

In this way, while creating new magazines, it will be possible to scroll down the available layouts (using the arrows on the tab “Add layout from repository”), see the preview of the structure and select one, using the button “Use this layout”.

Opencontent SCARL

Via Grazioli 106, 38122 Trento, Italy – P.IVA: 02190640223

Phone: +39 04611862944

Fax: +39 04611862844

E-mail: info@opencontent.it

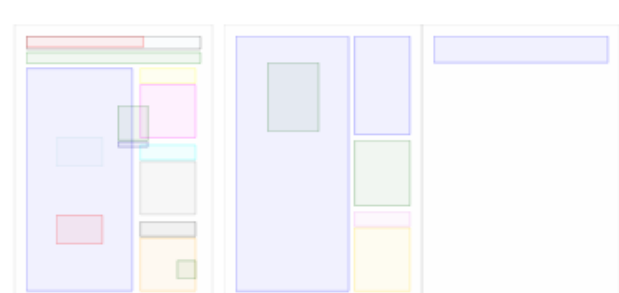
Web: www.opencontent.it

Name (required) :

Layout :

Griglia1 (1/9)

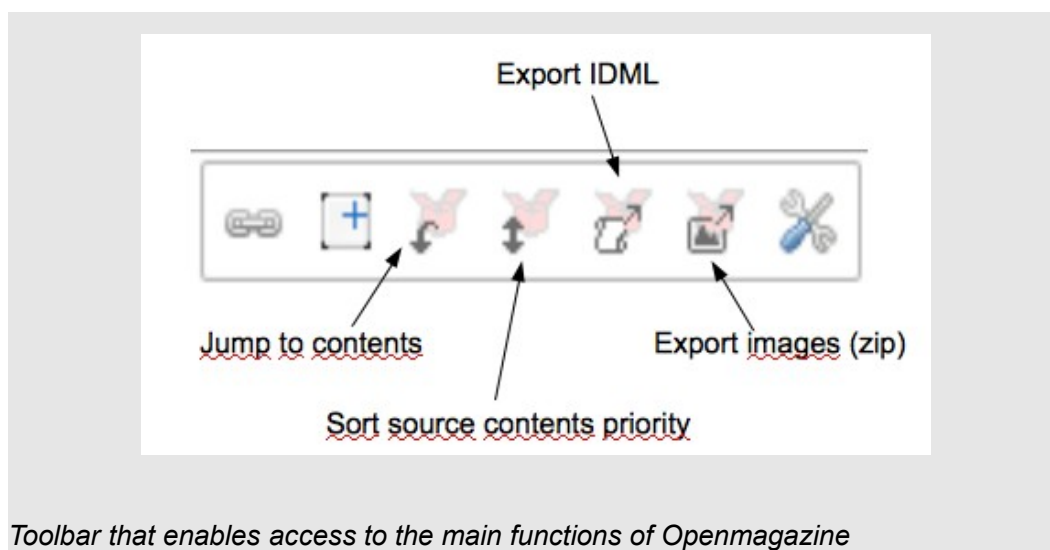
pages: 3
 contents: 5
 images: 4
 classes: article



☒ Insert eZPublish content from **Events**

From the tab “Add layout from repository” it’s possible to consult the list of filed layouts and select the chosen one

The extension uses the eZ Publish version management system (versioning). OpenMagazine also makes a toolbar available (see the following picture) that appears when the “magazine container” class is visible.



Opencontent SCARL

Via Grazioli 106, 38122 Trento, Italy – P.IVA: 02190640223

Phone: +39 04611862944

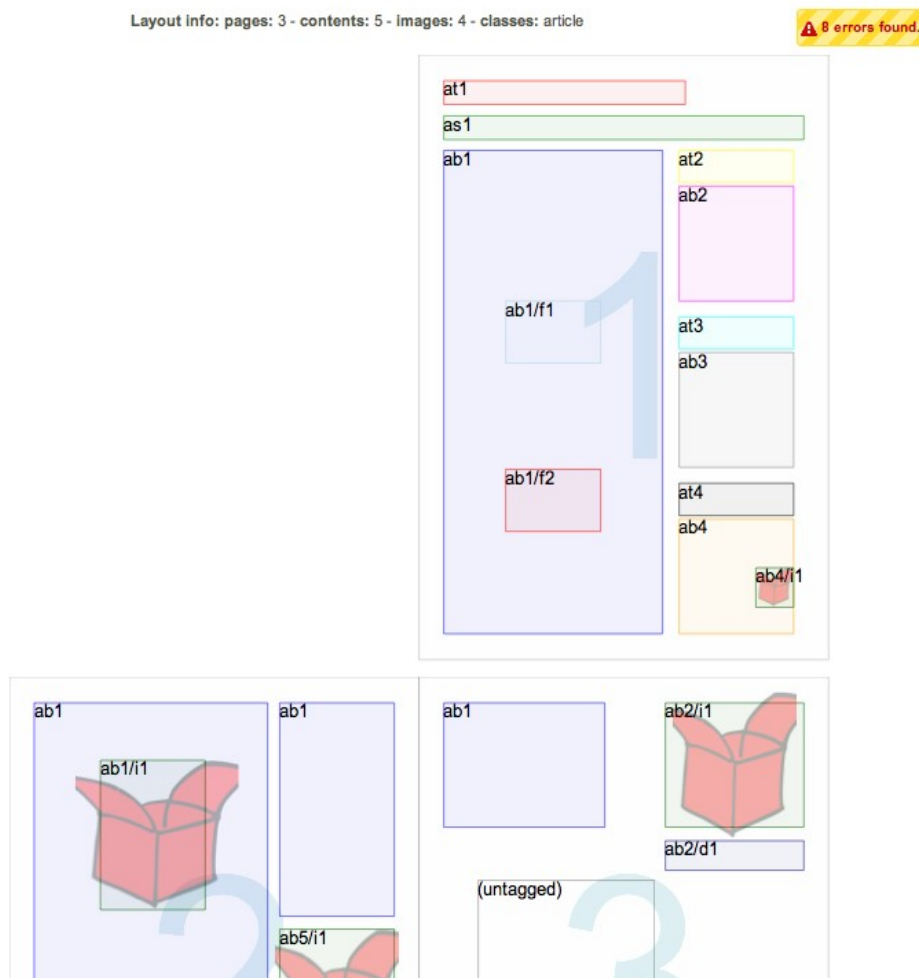
Fax: +39 04611862844

E-mail: info@opencontent.it

Web: www.opencontent.it

Laying out the magazine and validation of the information

Once the IDML file is loaded onto eZ Publish and the object “magazine section” has been published, the user can see a schematic representation of the graphic structure of the section. The on-line rendering is realised through the SVG format: in order to represent it correctly it is necessary to have an adequate browser (Firefox, Chrome, Safari).



OpenMagazine recognises all the lines of text and picture blocks defined in the IDML file; if these blocks have been tagged, the tag name appears inside the box (e.g. at1, as1, ...), if they haven't been tagged a warning appears (*untagged*). This representation enables the user to visually identify possible mistakes made during the tag mapping phase and to intervene by uploading a correct IDML.

When the mouse pointer is positioned over the text blocks, a tooltip appears with a preview of the text itself.

The contents required by the graphic structure are summed up in a single line (**Layout info**) (in the example, 5 “article” contents containing 4 pictures are requested); and an initial feedback on the final validation also appears (“no error found” or “number of errors found”).

In the part under the grid, for each article, the level of correspondence between the IDML file structure and the eZ Publish contents is shown, indicating possible incompleteness or mistakes;

Opencontent SCARL

Via Grazioli 106, 38122 Trento, Italy – P.IVA: 02190640223

Phone: +39 04611862944

Fax: +39 04611862844

E-mail: info@opencontent.it

Web: www.opencontent.it

in particular:

- for the **text boxes**, checks are carried out on the lengths of the texts. The bulks required for them (number of characters) are calculated when the IDML file is uploaded onto the platform, and in this phase they are compared with the contents actually in eZ Publish, according to a formula that allows for a percentage of tolerance established by the user in the configuration file *openmagazine.ini*, that can vary according to the attribute (title, sub head, etc).
- for the **picture boxes** it verifies the presence or lack of a picture to be inserted in the box.

Diagram illustrating the content validation process for an article.

Edit article

Content imported from **Petruciani biography** [expand all]

1. Body and Soul

2. Musica dentro intervista a michael radford

Expand all the information boxes

Tag	Attribute Identifier	Original Length	New Content Length	Contents validation
at2	title	42	42	$\text{Abs}(4400 - 4401) < 4401 * \text{tolerance\%}$ <p>Defined in openmagazine.ini</p>
ab2	body	4401	4400	
ab2/i1	body/image-1			
Tags imported from the IDML files		Text's lenght of contents from IDML imported files	Text's lenght of contents from eZ Publish	

For each tag, the length of the texts and the presence of pictures is checked; the user is warned when discrepancies are found between the IDML model and the eZ Publish contents

IDML file creation with contents from eZ publish

OpenMagazine is able to create a new IDML file that contains all the styles of the previously uploaded file and the contents extracted from eZ Publish.

Different methods are available for exporting the IDML files:

- through a web interface, exporting an IDML-type file and the pictures in zip format
- schedule operations with eZTransfer, that works with a standalone tool (provided with OpenMagazine).

Using OpenMagazine in an editorial context

As examples two possible methods for using OpenMagazine in an organisation are described.

Opencontent SCARL

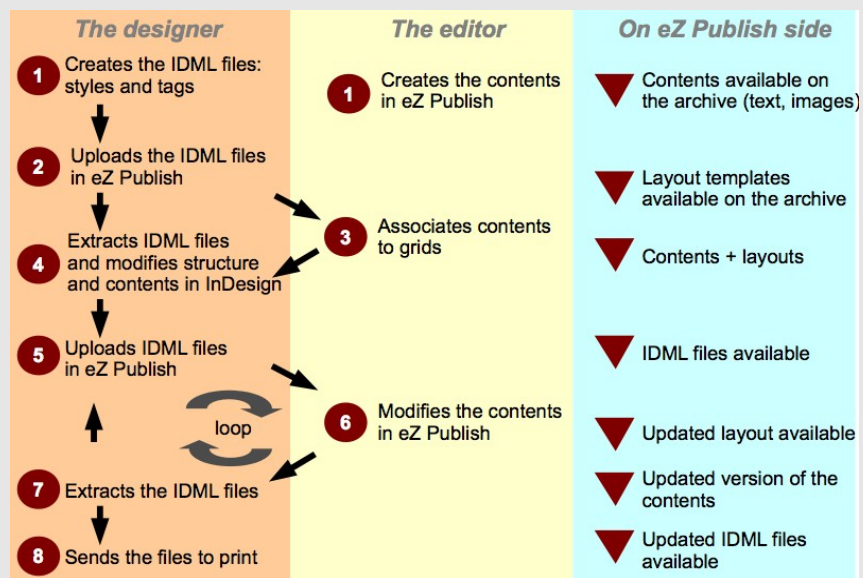
Via Grazioli 106, 38122 Trento, Italy – P.IVA: 02190640223

Phone: +39 04611862944

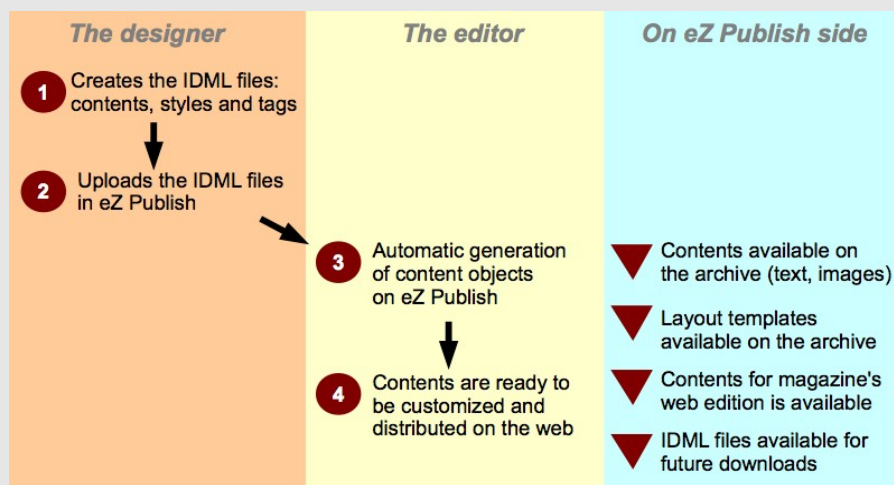
Fax: +39 04611862844

E-mail: info@opencontent.it

Web: www.opencontent.it



Case 1: the organisation has to manage a monthly magazine, editing and personalising each graphic detail. In this case, there will have to be continuous interaction between layout artists and journalists (loop).



Case 2: the organisation manages a daily news provider. It is probable that pre-established graphic models already exist, thus it may be enough to upload the contents onto eZ Publish in the already existing graphic models.

eZTransfer

With OpenMagazine the eZTransfer.jar executable file is provided (located in bin/java/). Running it from one's own pc, the user must download locally all the contents elaborated by OpenMagazine (this is performed by using the REST protocol) in IDML or XTG formats, besides the high resolution pictures, in order to enable the user to lay out the magazine and then send it to be printed.

Opencontent SCARL

Via Grazioli 106, 38122 Trento, Italy – P.IVA: 02190640223

Phone: +39 04611862944

Fax: +39 04611862844

E-mail: info@opencontent.it

Web: www.opencontent.it

The tool, realised with java technology and thus multiplatform, has been implemented in order to download the contents needed for layout and printing, without having to link up with eZ Publish.

eZTransfer is simple to configure: it requires the URL of the file in eZ Publish, a valid user for the *login*, the format with which to download the information (currently two formats are available: IDML for InDesign and XTG for Quark Xpress).

In order to customize the configuration, just uncompress the file eZTransfer.jar and edit the file:

```
ez/it/opencontent/eztransfer/res/ezprop.properties
```

An example of configuration:

```
base_url=http://openmagazine.opencontent.it/eng
login_url=http://openmagazine.opencontent.it/eng/user/login
listing_url=http://openmagazine.opencontent.it/eng/openmagazine/list/0/
username=admin
password=password
local_destination_path=my_local_folder
download_type=xpress
```

As “download_type”, you can choose between “xpress” (to generate XTG files) or “indesign” (to generate IDML files).

Configuration of the system

Please refer to the INSTALL file in the extension itself in order to install the OpenMagazine extension software package.

The main changes are made within the *openmagazine.ini* e *idml.ini* files. As well as the extension software, a multiplatform java tool is included – *eZTransfer*. This can be put into use by the publisher in order to download speedily into the publisher’s own file system the files generated by OpenMagazine (IDML and XTG, with the related images).

Requirements

Dependencies: ezmultipload, ezwebin, ezfind, ezwt, ezjscore

Requirements: eZ Publish 4.5 + with eZ Components; browser able to interpret the SVG format (Firefox, Safari, Chrome); Adobe InDesign CS5.

Packages with the extension

The openmagazine_classes_x.ezpkg package provides two classes: “magazine container” and “magazine section”. The “magazine section” class includes the “ezidml” data type; the “magazine container” class is a recipient, whose function is to group together and manage the individual sections of the magazine (page planning).

Configuration files: openmagazine.ini

[ContentTagMatch]

It is possible to define the rules for associating the contents of a sub-tree of eZ Publish (i.e. objects of any type and their related attributes) to the structure of the tags contained in the IDML file.

Group [ContentTagMatch]
Variables **Class[], Attribute[] and XMLTag[]**

The OpenMagazine XML tags must be labelled with:

- two letters of the alphabet and one numeral (e.g. *at1*)
- one letter and one number (e.g. *f1*).

Opencontent SCARL

Via Grazioli 106, 38122 Trento, Italy – P.IVA: 02190640223

Phone: +39 04611862944

Fax: +39 04611862844

E-mail: info@opencontent.it

Web: www.opencontent.it

The tags labelled by **two letters and one number** (such as *at1*) are first level tags and are therefore direct child tags of the root of the xml document.

In these tags:

- the first character represents an abbreviation of the class identifier defined in the rule describing the Class[] (e.g. Class[a]=article)
- the second character represents an abbreviation of the attribute identifier defined in the rule describing the Attribute[] (e.g. Attribute[t]=title)
- the digit represents its order of priority in the content tree (e.g. 1 = the first article according to the container's priority)

The tags defined by **one letter of the alphabet and one number** (e.g. *f1*) are children of first level tags that represent an attribute of the ezxmldata type.

Using these tags it is possible to unpack the ezxmldata so that they may be represented separately in the graphical grid.

In these tags:

- the letter of the alphabet represents the abbreviation of the ezxmldata custom identifier defined in the XMLTag[] rule (e.g. XMLTag[f]=factbox) ,
- the number represents where the custom tag occurs within the content of the ezxmldata (e.g. 1 = the first factbox inserted in the text)

Group [ContentTagMatch] Variable AttributeHandler[]

Each type of attribute can have a php class associated with it, which handles the export or the import of the specific attribute.

The rule requires the attribute identifier to be the key and the name of the php class to be the value: e.g. AttributeHandler[foo]=eZIdmlCustomFoo

The php class implements the eZIdmlCustomInterface.

By default the handler eZIdmlCustomEzXmlText is applied by attributes of the type ezxmldata: for this reason it is unnecessary to specify them.

Group [ContentTagMatch] Variable XMLTagHandler[]

Each type of custom ezxmldata may have a php class which handles the export or import of the specific custom ezxmldata.

The rule requires the ezxmldata identifier to be the key and the name of the php class to be the value: e.g. AttributeHandler[alternative_text]=eZIdmlCustomAlternativeText

The php class implements the eZIdmlCustomInterface

Group [ContentTagMatch] Variable DefaultCharLengthRatio=50

In this rule the default margin of error is defined for the relationship between the length of the original characters and the length of those imported by the contents of eZ.

The default value is 50. The formula used to calculate whether there is an error in the comparison between the character lengths is:

$$\text{AbsoluteValue} (100 * \text{length of original characters} - \text{length of imported characters}) / \text{original character length} > \text{CharLengthRatio}$$

If the formula result is greater than zero, the system produces an error which is displayable in the node view.

It is possible to define a specific margin of error for every attribute or custom ezxmldata by specifying in the key the value defined in the Attribute[] and in the XMLTag[]

Opencontent SCARL

Via Grazioli 106, 38122 Trento, Italy – P.IVA: 02190640223

Phone: +39 04611862944

Fax: +39 04611862844

E-mail: info@opencontent.it

Web: www.opencontent.it

E.g.:

```
CharLengthRatio[title]=20
```

```
CharLengthRatio[body/alternative_text]=15
```

[ExportImagesSettings]

Using this configuration line it is possible to resolve the problem of the synchronisation of images between the eZ filesystem and the local filesystem.

Because the IDML file registers absolute image file paths it is suggested to share a folder on the web server with the local filesystem (using webdav, NFS, ...).

If it is not possible to have a shared folder, it is possible to enable the download of a zip file containing the images and to specify a file path to a local folder from which to extract the zip file. See also: idml.ini/[ImageImport].

Group [ExportImagesSettings]

Variable DownloadZip

If the value is 'enabled' in the OpenMagazine website toolbar a button will appear through which it is possible to download a zip file of the pictures related to the relevant section.

Group [ExportImagesSettings]

Variable LanImagePath

Absolute path of the shared folder in the web server.

Group [ExportImagesSettings]

Variable LocalImagePath

Absolute path of the picture folder in the local filesystem.

[ExportSettings] and [ExportClassSettings]

Through these configurations it is possible to define the types of exports.

In JSON the system shows the list of class nodes defined in

ExportClassSettings/MagazineContainerClassIdentifier downloadable at the following address:

`http://<site_url_with_siteaccess>/openmagazine/list/0/<export_handler>`

In this list there are the addresses for exporting each class node defined in

ExportClassSettings/MagazineSectionClassIdentifier and the related pictures in the format defined by the export_handler.

The available values are *indesign* and *xpress*.

The indesign handler shows the IDML files of the selected magazine and the ZIP files of the related pictures.

The xpress handler shows the XTG files for each article of the selected magazine and the files of the related pictures.

Group [ExportSettings]

Variable DefaultHandler

API of default export: the default handler is used to create the list of class nodes defined in the available ExportClassSettings/MagazineContainerClassIdentifier and as a fallback in the case of a non-existent export_handler.

Group [ExportSettings]

Variable ExportHandlers[]

Export handlers for indesign and xpress.

Opencontent SCARL

Via Grazioli 106, 38122 Trento, Italy – P.IVA: 02190640223

Phone: +39 04611862944

Fax: +39 04611862844

E-mail: info@opencontent.it

Web: www.opencontent.it

Group [ExportClassSettings]**Variable MagazineParentNode**

Node ID of the exportable node container: the default is node 2.

The system shows all the nodes defined in ExportClassSettings/MagazineContainerClassIdentifier descending from the node defined here.

Group [ExportClassSettings]**Variable MagazineSectionClassIdentifier[]**

Here are defined the 'container' classes accepted for export.

Group [ExportClassSettings]**Variable MagazineContainerClassIdentifier[]**

Here are defined the 'section' classes accepted for export.

Configuration files: idml.ini

Group [FileSettings]**Variable PublishedIdml**

Name of the folder in the web server where the IDML files imported into eZ are saved.

Group [ClassSettings]**Variables Container[], Section[]**

The classes contained in the array Section are the ones that manage the datatype ezidml.

The classes contained in the Container are the container-type classes.

Group [FileImport]**Variables DefaultImportClass, DefaultImportNameAttribute, DefaultImportIdmlAttribute**

Default classes and attributes for importing the IDML files through eZMultiupload.

Group [ImageImport]**Variable ImportImagePath**

Web server folder path from which the system imports pictures during creation or update of the contents starting from an IDML file.

Group [Repository]**Variables UseRepository, ParentNode**

Qualification and definition of the Repository IDML node.

Licence

OpenMagazine is distributed under the eZ Proprietary Extension License (PEL), Version 1.3

About us

OpenContent develops Enterprise Content Management applications for Digital Media and Public Administration since 2004. Opencontent has developed significant projects at national and international level, continuously integrating into its solutions the innovation which comes from the industrial research (as FBK – Human Language Technology) and its business partners (as eZ Systems), with the aim of offering competitive and reliable services.

Opencontent SCARL

Via Grazioli 106, 38122 Trento, Italy – P.IVA: 02190640223

Phone: +39 04611862944

Fax: +39 04611862844

E-mail: info@opencontent.it

Web: www.opencontent.it

Some services we provide include:

- content management strategy and consultancy
- business analysis for digital media
- semantic software, which automatically analyzes and understands the contents and the connections in your information sets (data mining)
- support services.

Contacts

OpenContent SCARL [free software solutions]

via Grazioli 106 - 38122 Trento (TN) Italy

E-mail: info@opencontent.it, acquisti@opencontent.it

Web : www.opencontent.it

Phone: +39 0461 1862944

Fax: +39 0461 1862844

Mobile: +39 340 6714485

Opencontent SCARL

Via Grazioli 106, 38122 Trento, Italy – P.IVA: 02190640223

Phone: +39 04611862944

Fax: +39 04611862844

E-mail: info@opencontent.it

Web: www.opencontent.it