

Table of contents

Configuration	1
Configuration	1
WebDAV setup	1
Locale Settings	4
eZ publish running on a CGI version of PHP	5
Path prefix	5
Introduction	6
Configuration files	7
Site access	8
Common settings	11
Multi Site	12
Directory structure	13
Language and charset	15
Cron jobs	25
Login handler	27
Search engine	31
Tips & Tricks	35
Security	42
Non-Virtualhost	42
Discussions	45
Restrictions	45
System security	46
Optimization	46
PHP Acceleration	46
Disabling the cache	47
Configuration tuning	48
Caching for improved speed	49
Speeding up acrobat pdf document indexing	49
Backup & Restore	50
Move to new provider	51
Overview	52
Troubleshooting	54
404 - Page not found	54
Change the right configuration file	54
Debug popup window infinite loops	55
eZ publish 3 and Apache 2	55
Getting looped back when logging in	56
Giving you back the admin rights	56
I need to reindex my site for search.	56
Images do not work	57
Mixing up node id and object id	59
Not enough PHP memory	59
Open_basedir restriction	60
Permissions of cache files	62
Problems with the translation function	62
Process cache enabled reduces performance	62
User site instead of admin site	63
Warning: Failed to write session data (user).	63
What not to do with eZ	64

Configuration

Here you will find information about how to configure eZ publish.

Configuration

Here you will find information about post install configuration.

WebDAV setup

This document explains how to set up and use the beta version of the eZ WebDAV content administration sub-system. The functionality described here is new and will appear in eZ publish 3.2 (also in beta releases).

A few words about WebDAV

WebDAV is an abbreviation for "Web-based Distributed Authoring and Versioning". It is a set of extensions to the HTTP protocol which allows users to collaboratively edit and manage files on remote web servers using WebDAV enabled clients. For example, you can use recent versions of KDE's Konqueror or Microsoft's Internet Explorer to browse/manage the content of a WebDAV compatible web server (similar to browsing/managing local filesystems). DAV is an IETF Proposed Standard (published as RFC 2518). This means that it is an entirely open standard.

eZ publish and WebDAV

The beta version of eZ publish 3.2 contains a WebDAV server library. This library provides a basic framework and takes care of the cumbersome low level communication between a WebDAV client and the server itself. In other words, people can use this library to implement their own servers very easily, since most of the dirty work is already done (and will be available under the GPL license).

In addition to the base library, there also exists a server (named eZWebDAVContentServer)

which can be used to remotely administer the actual content of an eZ publish site. This means that from now on, it will be possible to browse and manage the content (folders, articles, images, files, etc.) in the same fashion as when browsing a local filesystem. If you're running a recent version of Windows or KDE, you'll be able to log in and manage your content in a matter of seconds. The browsers of these systems already have built-in WebDAV support. There is no need to worry about firewalls since WebDAV is plain bi-directional HTTP traffic. If you are able to browse/surf the site from your workstation, you can use WebDAV to administer it.

Features

The WebDAV subsystem has just been implemented, it is therefore a bit limited. The user accessing the WebDAV interface must have administrator privileges in order to be able to browse and administer the content. For now, only the following operations are supported:

- Content browsing (navigation)
- File upload
- File download
- Folder creation
- Renaming

Content browsing

The WebDAV content server provides an interface for browsing the content of the site. Folders are displayed as folders/directories while binary files and images are displayed as regular files. For now / by default, everything else is displayed as folders.

File upload

From within a WebDAV enabled client running on some workstation, any kind of file may be uploaded to the server. If an image is uploaded, the content server subsystem will recognize it as an image and will automatically create an ezimage object. All other filetypes will be stored as ezbinary files.

File download

It is possible to download two things from the server to the local filesystem of the host running the WebDAV client: images and binary files.

Folder creation

It is possible to create folders. These will be created as regular folders within the site.

Folder/file renaming

It is possible to rename files and folders. There are currently some bugs in this feature. The second or third, and the following rename operations usually fail. This is due to a database issue that will be fixed later on.

Some of functionality mentioned above may be configured from within the WebDAV settings file (settings/webdav.ini).

Possible future features:

Copy operation (copy things from one location to another)

Move operation (move things from one location to another)

Delete operation (recursive removal of stuff)
Attribute manipulation (manipulation of certain attributes)
Multi user support (support for multiple users, with permissions, etc.)
Support & automatic parsing/processing + storage of common/various filetypes.

Setting it up

First of all, please keep in mind that the WebDAV subsystem of eZ publish is still in the beta stage, it may not work correctly with every client. Secondly, please be patient and follow the setup instructions carefully, step-by-step. The WebDAV server has only been tested on the Linux platform, running Apache 1.3, PHP 4.3.2 and MySQL 4.1.0-alpha. Clients used for testing during development: various versions of Internet Explorer (mainly under Win2K and XP), the latest version of KDE's konqueror and a text mode client called nd. The following five steps describe how to set up the WebDAV content server.

(1) Enabling WebDAV functionality

The WebDAV functionality has to be enabled. Bring up the settings/webdav.ini file in your favourite editor and change the EnableWebDAV setting from false to true.

(2) DNS entry (optional)

Set up a subdomain as an alias for the hostname you use to access the site. For example, you could use "webdav.yoursite.net", which will have to point to the same IP address as www.yoursite.net. As the headline points out, this is optional - it isn't needed if you only want to test & see how the WebDAV system works. If omitted, you should simply replace "webdav.yoursite.net" with only "yoursite.net" in the following steps. You will still be able to test the WebDAV interface, however - the rewrite rule may mess up normal site browsing.

(3) Apache configuration

There is a "webdav.php" file in the root directory of the eZ publish installation. This is the file that provides the WebDAV interface to the site content. The web server has to automatically execute this file whenever a WebDAV client issues a command to the server (browsing, upload, download, etc.). A virtual host section in httpd.conf should take care of this (look at the example below).

```
NameVirtualHost IP-Address

<Virtualhost IP-Address>
  <Directory /path/to/the/root/of/your/ezpublish/installation>
    Options FollowSymLinks Indexes ExecCGI
    AllowOverride None
  </Directory>
  RewriteEngine On
  RewriteRule !\.(css|jar|js|html|php)$ /webdav.php
  ServerAdmin root@localhost
  ServerName webdav.yoursite.net
</VirtualHost>
```

If everything is running locally: it is still possible to set up and test WebDAV. In this case, the following line should be added to the local hosts file: "xxx.xxx.xxx.xxx webdav", where xxx.xxx.xxx.xxx is the real IP address of the computer. In addition, the ServerName setting in the

VirtualHost declaration (see above) should be set to "webdav".

(4) Testing

Fire up a WebDAV enabled client and try to browse the site(s) using the WebDAV-URL.

Windows:

Should work okay from within Win2K and XP. However, sometimes it just fails under XP (will be fixed later). Make sure you have a recent version of Internet Explorer (6.0.2800.1106 or above) and attempt to connect to a so-called "Web Folder" (either from within IE's file menu - > open and check the "open as webfolder" box; or use "My Network Places"). Try to connect/browse using the following syntax: *http://webdav.yoursite.net/*

If everything is running on one computer (testing it locally with "xxx.xxx.xxx.xxx webdav" added to the local host file), the URL would be the following: *http://webdav/*

KDE/Konqueror:

Make sure you have a recent version of Konqueror (preferably 3.1.3 or later). Attempt to browse the WebDAV server using the following syntax: *webdav://webdav.yoursite.net/*

If everything is running on one computer (testing it locally with "xxx.xxx.xxx.xxx webdav" added to the local host file), the URL would be the following: *webdav://webdav/*

Please remember that you will need to log in with an eZ publish user who has administrator privileges.

Locale Settings

The locale is set to:

site.ini:RegionalSettings:SystemLocale is set in case it is not empty, otherwise the system locale is used.

For templates this default locale is overridden with:

template.ini:CharsetSettings:DefaultTemplateLocale if it is not empty a `{*?template locale='...'?}` setting in a template, in case it exists.

All settings may consist of a comma separated list. eZ publish will try to set the locales in order as they appear in the array until one of the settings works.

Example:

```
site.ini:RegionalSettings:SystemLocale=posix,C
template.ini:CharsetSettings:DefaultTemplateLocale=farawaistan
foobar.tpl has {*?template locale="german,de_DE"?}
```

the list which eZ publish tries to set for the whole system is:

posix, C

the list which eZ publish tries to set for a template is:

german, de_DE, farawaistan

If none of the locales in the last line is supported on the system, the one that works from the 'global' list is still in effect. It is advised to leave `template.ini:CharsetSettings:DefaultTemplateLocale` set to the empty string if NO other template overrides the locale. There is no overhead of setting locales at all then. In that case there is a template which overrides the locale, make sure to set a default locale for other templates, otherwise templates will inherit the locale if they are called from within that template.

eZ publish running on a CGI version of PHP

only available in 3.4 or higher

If PHP is not running as an Apache module eZ publish would not run before version 3.4. From version 3.4 this now works without problems, but with one small difference. As CGI does not accept path-info like:

```
http://example.com/site/articles/new
```

or:

```
http://example.com/site/index.php/articles/new
```

Instead it wants this:

```
http://example.com/site/index.php?/articles.new
```

eZ publish handles this for you, so there is no problem here either. The only thing you'll need to run eZ publish with CGI is a standard configuration of PHP running on CGI, something like this:

```
ScriptAlias /php/ /usr/local/bin/  
AddType application/x-httpd-php .php  
Action application/x-httpd-php /php/php-cgi
```

(PHP's CGI module is installed as `/usr/local/bin/php-cgi` here).

Because of performance reasons it's still advised to use the Apache module version of PHP if you have the possibility to do so.

Path prefix

If we have more site access used to display different part of our content tree, we often want to remove parts of the url alias (because this part will be a duplicate of the host). This can be solved with a Path prefix.

Example

Our eZ publish installation have two folders placed in content. One called site1 and the other called site2. Our first site access should only access content in the site1 folder, and the second in site2. This is easy to set up with permissions, but the url aliases (the nice URLs dynamically generated based on the tree structure) will default include the name of the site1 or site2 folder in the url. This makes the URLs not as nice as they could be.

URL's without this functionality:

```
http://www.site1.com/site1/en/my_article  
http://www.site2.com/site2/no/another_article
```

If we wish to improve this, we configure like this:

in settings/siteaccess/site1/site.ini.append:

```
[SiteAccessSettings]  
PathPrefix=site1
```

and in settings/siteaccess/site2/site.ini.append:

```
[SiteAccessSettings]  
PathPrefix=site2
```

URLs will then be like this:

```
http://www.site1.com/en/my_article  
http://www.site2.com/no/another_article
```

This prefix will then be removed when we request `$node.url_alias` to get the url alias of a node. The standard url's (`/content/view/full/42`) will still work as normal.

Introduction

Configuration files

eZ publish can be configured with a set of configuration files. Configuration files are placed in the settings/ directory in the eZ publish root directory on the server. The main configuration file is called site.ini and is located directly in the settings/ directory. site.ini is divided up in several blocks. A block name is placed inside brackets.

Example:

```
[DatabaseSettings]
DatabasePluginPath=
DatabaseImplementation=ezmysql
Server=localhost
User=publishuser
Password=publishpass
Database=publishdb
UseSlaveServer=disabled
ConnectRetries=0
Charset=iso-8859-1
UseBuiltinEncoding=true
Socket=disabled
SQLOutput=enabled
UsePersistentConnection=disabled

[RoleSettings]
EnableCaching=true
PolicyOmitList[]=user
PolicyOmitList[]=layout
PolicyOmitList[]=manual
PolicyOmitList[]=ezinfo
```

In this example we have two blocks, DatabaseSettings and RoleSettings. Each block have several variables that decide the behaviour of our site. Some of the variables has a name ending with [], this means that the value is treated as an array, and we can add as menu values to this variable as we want.

Error???

Configuration files

Overriding site.ini

If we edit directly into site.ini we might have several problems if we wish to upgrade to a new version later on. The file might be overwritten by the new install, or if we take a backup of our file first, we might loose new settings included in the newer release.

To avoid these problems we have an override system for the configuration file. In settings/override/ we create a file called site.ini.append or site.ini.append.php. You should always use site.ini.append.php and place the variables inside PHP-comments (*/** and **/*) on a non-virtual host setup for security reasons.

All variables placed in this file would override the variables placed in the main site.ini file.

Typical content of settings/override/site.ini.append:

```
[DatabaseSettings]
User=nextgen
Password=nextgen
Database=nextgen

[SiteAccessSettings]
HostMatchType=map
HostMatchMapItems[ ]=www.johenrik.com;user
HostMatchMapItems[ ]=admin.johenrik.com;admin
MatchOrder=host
```

This configuration file sets the database settings and site access settings.

Configuration files

In the /settings directory you will find several files and directories.

The *.ini files directly under the directory settings/ are default configuration files and should **not** be changed. Use them to look up some explanations about configuration options, but don't change them.

The files in /settings/override can be used to override configuration parameters. They are **always** valid and override any settings from other configuration files. These are global settings.

The directories in /settings/siteaccess are for different siteaccess rules. E.g. /settings/siteaccess/user is for the user site while /settings/siteaccess/admin is for the admin site. The files in /settings/siteaccess/[directory] are special settings that only count for that siteaccess method (e.g. a certain language or a different design). They get overridden by the /settings/override parameters, but they override the default parameters.

The configuration files are read in this order:

```
/settings/site.ini
/settings/siteaccess/[directory]/site.ini.append
/settings/override/site.ini.append.php
```

This means that settings in [3] override [2] which override [1]. If you want to change configuration settings for different siteaccess setups then create new directories under /settings/siteaccess and change them there (also remember to put those new siteaccess rules in /settings/override/site.ini.append.php).

It sounds complicated, but it's actually easy, logical, and makes sense. It also makes upgrades simpler, if you stick to this.

Site access

Once you've gotten eZ publish up and running on a server you need to configure the system. You can have several different sites running on the same eZ publish installation. To distinguish between these sites you need to set up something called site access. The site access defines how eZ publish will recognize which site you're accessing. eZ publish will then know which database to use, which design to show etc. For normal, small sites we have usually two site accesses, the user site and the admin site.

Host

In this example we will name our site `www.example.com` and we will use `admin.example.com` as the administration interface. To make eZ publish fetch site access from host names you need to configure a DNS server and point the domains to your web server. When your DNS is up and running and the names resolve to your web server and your eZ publish installation you need to make eZ publish recognize the names and use the correct configuration. To do this you add a section called `[SiteAccessSettings]` in your configuration file.

`settings/override/site.ini.append:`

```
[SiteAccessSettings]
MatchOrder=host
HostMatchRegex=^(.+)\.example\.com$
HostMatchSubtextPost=\.example\.com
```

Here we told eZ publish to take the part before `.example.com` of the url and map it to a directory in `settings/siteaccess/`. If we enter `www.example.com` in our browser eZ publish will look a directory called `www` in `settings/siteaccess/`. Now that eZ publish knows how to distinguish between the two domains we need to create a configuration file for each site. This is done by creating two folders under `settings/siteaccess/` which corresponds to our matches (`www` and `admin`). In both these folder you need to create a file called `site.ini.append`. This is the configuration file which will be used to override any of the standard settings in eZ publish just like `settings/override/site.ini.append`, but will only work for the current site access. We will keep our example simple and have just made a few settings to distinguish between the two sites. You can see the two configuration files below.

`settings/siteaccess/admin/site.ini.append:`

```
[SiteSettings]
LoginPage=custom

[SiteAccessSettings]
RequireUserLogin=true
```

The configuration `LoginPage=custom` means that eZ publish will use a separate template for the login page of the administration site. `RequireUserLogin=true` tells eZ publish not to let anyone inside eZ publish unless they're logged into the system.

`settings/siteaccess/www/site.ini.append:`

```
[DesignSettings]
SiteDesign=mydesign

[SiteAccessSettings]
```

```
RequireUserLogin=false
```

The settings above applies to the website. SiteDesign=mydesign means that eZ publish will prefer to use the design for the site found in design/mydesign/, RequireUserLogin=false is set so that users do not have to log into eZ publish to browse the website.

Host Match Map

In the previous example we used a regexp (regular expression) to map part of the url to different sites. This is a very powerful way of matching, but if you do not have experience or do not understand regexp, this approach might be difficult. Instead of regexp we can make a list of URL's and tell eZ publish 3 to match them to a site. See the configuration example below. settings/override/site.ini.append:

```
[SiteAccessSettings]
MatchOrder=host
HostMatchType=map
HostMatchMapItems[ ]=example.com;user
HostMatchMapItems[ ]=www.example.com;user
HostMatchMapItems[ ]=admin.example.com;admin
```

As you might see from the example we mapped example.com and www.example.com to the user site and admin.example.com to the admin site. To make individual settings for the two sites edit settings/siteaccess/user/site.ini.append and settings/siteaccess/admin/site.ini.append.

URI

We will stick with the www.example.com, but now we will use URI to recognize the different sites. http://www.example.com/index.php/admin will be our URL to the admin site and http://www.example.com/index.php the URL to our user site. Here we only need the www.example.com domain to point to our web server. Set these settings in our configuration file:

```
[SiteSettings]
DefaultAccess=user

[SiteAccessSettings]
MatchOrder=uri
```

With the DefaultAccess variable we told eZ publish to use the user site if it is unable to match a site. Now eZ publish will take the last part of the entered URL and map it do a directory under settings/siteaccess/. If we now enter http://www.example.com/index.php/admin, eZ publish will look for a directory called admin. Using URI is useful when you want multiple sites but don't have a domain for each site or don't have the ability to setup virtual hosts for each site. This is the configuration the eZ publish Windows installer uses.

Port

By setting up site access to use ports we can let eZ publish distinguish different sites by mapping a port to a site. In this example we will use port 80 for the user site and port 81 for the admin site. When the configuration is done we should be able to access the user site on `http://www.example.com:80` and the admin site with `http://www.example.com:81`. The settings in our configuration file for this setup is shown below.

```
[SiteAccessSettings]
MatchOrder=port

[PortAccessSettings]
80=user
81=admin
```

Index

If all others fail, index parameters is one of the last options.

With the above settings, any addition to `index_` 'll be used for access purposes. `index_admin.php` for admin, `index_users.php` for users. To use this, copy `index.php` to `index_admin.php` and `index_users.php` and rename

```
SiteURL=www.domain.tld/index.php
AdminSiteURL=www.domain.tld/index.php/admin
```

to

```
SiteURL=www.domain.tld/index_users.php
AdminSiteURL=www.domain.tld/index_admin.php
```

Unix: The best way to copy `index.php` is by creating a symbolic link: `ln -s index.php index_admin.php`. This way any change to `index.php` will be copied to all symbolic linked files.

Common settings

Often used settings

DatabaseSettings

User	Username for connection to database
Password	Password used for connection to database
Database	Name of database used by eZ publish

SiteAccessSettings

HostMatchType	How we separate different site accesses. Valid values are: disabled, map, element, text and regexp
HostMatchMapItems[]	When using map as HostMatchType, this is used to define each site access
MatchOrder	An array of access types which are tried, possible entries are: host, index, uri and port
RequireUserLogin	Do the user have to login to use the site? This setting is usually placed in the site.ini.append for the specific site access

TemplateSettings

Debug	If this is enabled the system will print out the name of each template file that is included in the processing of the page
-------	--

DebugSettings

DebugByIP	Whether debug is set per IP address. If set to enabled it will use DebugIPList for valid IPs otherwise all IPs get debug
DebugIPList[]	An array with IP addresses which will get debug information
DebugOutput	Master switch for debug, if disabled no debug will be shown. If enabled it's up to the DebugIP and DebugIPList to decide what to do
Debug	Controls how debug is displayed. Valid values are: inline and popup
DebugRedirection	Controls if redirects should be debugable, set to enable to get a redirect page with debug info

ContentSettings

ViewCaching	Whether to use view caching or not. Should be disabled on development sites
-------------	---

DesignSettings

SiteDesign	The design on the site. This setting is usually placed in the site.ini.append for the specific site access
------------	--

Multi Site

If you are setting up several sites with one eZ publish 3 installation using site access, you might want to separate the cache and storage directory. This will make things less cluttered and easy to manage when having many sites per installation.

The default settings in eZ publish 3 stores cache and files in a common directory for all sites. This can cause problems if you e.g. want to move one of the sites to another server in the future. This can be changed by setting some variables in your site.ini.append for each of your sites.

In this example we will separate storage files and cache for 'myfirst.ezpsite.com', 'admin.myfirst.ezpsite.com', 'mysecond.ezpsite.com' and 'admin.mysecond.ezpsite.com'. After setting up site access (which you can read more about [here](#)) we need to set some additional settings in the site.ini.append files.

```
file: settings/siteaccess/myfirst/site.ini.append
[FileSettings]
StorageDir=myfirst/storage
CacheDir=myfirst/cache
```

```
file: settings/siteaccess/admin.myfirst/site.ini.append
[FileSettings]
StorageDir=myfirst/storage
CacheDir=myfirst/cache
```

```
file: settings/siteaccess/mysecond/site.ini.append
[FileSettings]
StorageDir=mysecond/storage
CacheDir=mysecond/cache
```

```
file: settings/siteaccess/admin.mysecond/site.ini.append
[FileSettings]
StorageDir=mysecond/storage
CacheDir=mysecond/cache
```

Now we have told eZ publish to store myfirsts cache in var/myfirst/cache and myfirsts files in var/mysecond/storage. The same goes for the mysecond site.

Important: It is important that both the user and the admin site has the same storage directory. Otherwise if you e.g. upload a new file using the admin site, the user site will not be able to locate the file because it is looking for it in the wrong directory.

Important: Make sure that Apache has sufficient rights for the folders you create.

Note: The cache for ini-files and translations can not be separated as described in this document

Directory structure

Directory structure explained

Should you not be familiar with standard terms such as content objects or nodes, please take a look at [Datatypes, Classes, Objects and Nodes](#) first.

The templates are stored in the folder **/design**. Here you will find the folder **/design/standard**, **/design/admin/**, **/design/user/** and **/design/demo/**, which each represent a global site design. The basic principle for creating custom designs is to create override templates in your custom design

folders that replace the designs in `/design/standard`. If eZ publish does not find an override template it simply uses the template in the standard directory.

In the current set-up the *admin* folder contains the overrides and functionality for the admin section, the *user* folder is a basically empty design based on *standard* and the *demo* folder contains the overrides and functionality for the demo site that comes with the installation. The latter is a good starting point to look through when you want to examine how to create your own design. Take a look at [Template Language](#) for further reference. Please note that template caching should be turned off while you develop or edit templates ([How do enable/disable the cache function?](#)).

Standard design folder

`/design/standard/fonts/` - font data for the TextToImage function
`/design/standard/images/` - the global images
`/design/standard/override/` - some specific templates that override those in `.../templates/` (below)
`/design/standard/prepare/` -
`/design/standard/stylesheets/` - CSS files
`/design/standard/templates/` - main templates folder

In this folder you find

`.../class/` - generic content class handling templates
`.../class/datatype/edit/` - specific datatype editing templates when creating/editing content classes. Only those that need specific handling are found here.
`.../content/datatype/edit/` - generic datatype editing templates for actual content objects. Here you find all datatypes.
`.../content/datatype/view/` - generic datatype viewing templates
Note: info on the above three directories in context at http://ez.no/sdk/kernel/view/content_datatypes
`.../content/datatype/view/xmltags/` - generic xml tags viewing templates
`.../content/view/` - generic content object viewing templates (e.g. for an article)
`.../node/` - generic node handling templates
`.../node/view/` - generic node viewing templates (e.g. for an article). Difference to `content/view` (above) is, that a node is a pointer to a content object (actual page) within the site hierarchy.
`.../error/kernel/` - standard error message templates

and many more.

Custom design folders (all folders except for the standard folder)

`/design/(yourdesign)/stylesheets/` - CSS files
`/design/(yourdesign)/images/` - custom images
`/design/(yourdesign)/templates/` - this is where all the generic templates of your design go. The main file will be the *pagelayout.tpl* that defines the default page design (equivalent of *frame.php* in the 2.x series). To make section specific page designs put a *pagelayout_section_[sectionnumber].tpl* in `/design/(yourdesign)/override/templates/` (see below).

/design/(yourdesign)/override/ - this is the top level directory for all the templates that override those that are in the standard design folder or in your generic design folder. This is where probably most of your custom designs go.

/design/(yourdesign)/override/templates/ - directory for section templates. These override */design/(yourdesign)/templates/pagelayout.tpl* (see above)

Example: /design/(yourdesign)/override/templates/pagelayout_section_[sectionnumber].tpl - override layout for section number [sectionnumber]. If you create a new design, these files are most important.

/design/(yourdesign)/override/templates/content/ - override templates for your handling (editing, creating) content objects (either self created or predefined objects) as well as datatypes

/design/(yourdesign)/override/templates/node/ - override templates for *viewing* content objects and pages. This is the second most important directory, where all custom content templates go (these complement the pagelayout templates from above)

Language and charset

eZ publish support many different charsets and languages.

Installing a Language pack

Sample german language update (ger-DE)

after demo install

go to http://www.ez.no/developer/ez_publish_3/translations and download the file *ezpublish-3.x-x-ger-DE.zip*

unpack the file

Place the *.ts* in the ez folder */share/translations/ger-DE*

Place the *.ini* in the ez folder */share/locale*

edit */settings/override/site.ini.append.php* or the correspondant files from the */settings/siteaccess* folder

```
[RegionalSettings]
Locale=ger-DE
ContentObjectLocale=eng-GB
TextTranslation=enabled
[ContentSettings]
TranslationList=ger-DE;eng-GB
```

Recommended: activate translation cache

```
[RegionalSettings]
Locale=ger-DE
ContentObjectLocale=eng-GB
TextTranslation=enabled
TranslationCache=enabled
[ContentSettings]
TranslationList=ger-DE;eng-GB
```

Clear the Cache
Reload site

If reloading takes more than 30 seconds, chances are that you have not enabled the mbstring extension in PHP. When this is enabled, translation cache regeneration should not take much more than 6-8 seconds.

If you don't have the PHP mbstring extension, then reloading may time out. The quickest way to fix this to give eZ publish more RAM and/or execution time. (The best fix is to enable mbstring.) See php.ini. I use 32MB and 300 sec on a AMD 800. The german language file is actually the largest of them all. It is hard to KISS in german ;-).

edit php.ini (Linux: /etc/php.ini Windows: C:\windows\php.ini)

```
;;;;;;;;;;;;;
; Resource Limits ;
;;;;;;;;;;;;;

max_execution_time = 300      ; Maximum execution time of each script,
in seconds
max_input_time = 180        ; Maximum amount of time each script may spend
parsing request data
memory_limit = 32M          ; Maximum amount of memory a script may consume
(8MB)
```

If you still notice untranslated phrases it might be that the information either comes out of the database or that this information is hardcoded. Otherwise try to clear und rebuild the cache.

Multilanguage site

One way of setting up a multilingual site is to create one siteaccess for every language. This siteaccess will then set the ContentObjectLocale you want to use for that specific site.

E.g.

```
ez.no/norwegian ( ContentObjectLocale=nor-NO )
ez.no/english ( ContentObjectLocale=eng-GB )
```

eZ publish will then prefer to show a norwegian translation on the norwegian site. If it does not

exist it will show the default translation.

To get the dropdown you only need to link to the same URL, but with a different URL prefix.

This setup works with a 1-1 translation. If you want different content on the different sites you need to build two different directory structures.

Unicode with eZ publish

PostgreSQL

To use unicode with eZ publish you should use PostgreSQL. MySQL has Unicode support in version 4.1.1 and newer, which is not stable at the time of writing this.

Create database

To be able to run eZ publish with Unicode (UTF-8) support you need to create a database which has support for this. This is done with the createdb command.

```
$ createdb -E UNICODE mydatabase
```

To list the available databases and see what kind of charsets they support you can run psql -l.

```
$ psql -l
      List of databases
  Name  | Owner  | Encoding
-----+-----+-----
nextgen | postgres | UNICODE
template0 | postgres | SQL_ASCII
template1 | postgres | SQL_ASCII
(3 rows)
```

MySQL

From version 4.1.1 MySQL has support for unicode. To convert your existing database to utf-8 do the following:

Stop mysql

```
mysqladmin -uroot -p<password> shutdown
```

You need to run this to convert your database

```
myisamchk -r -q --set-character-set=utf-8  
/path/to/mysql_data_files/database/ez*.MYI
```

Example

```
myisamchk -r -q --set-character-set=utf-8  
/var/lib/mysql/nextgen/ez*.MYI
```

If you create a new database you can (from MySQL version 4.1.1) use:

```
create database nextgen character set utf8;
```

Setting charsets in eZ publish

To make eZ publish handle Unicode properly you need to configure which charsets the different layers use. You will need to configure database, internal locale and default template charset. This is configured in the files:

site.ini
i18n.ini
template.ini

```
site.ini:  
[DatabaseSettings]  
Charset=utf-8
```

```
i18n.ini:  
[CharacterSettings]  
Charset=utf-8
```

```
template.ini:  
[CharsetSettings]  
DefaultTemplateCharset=utf-8
```

That's it. You can now enjoy Unicode characters with eZ publish.

Known problems

The search engine does not properly index languages which does not use space as word separator, like chinese and japanese. **Update:** The current unstable (svn) version has support for this. It will be standard from version 3.2.

Creating a new translation

What you need

eZ publish 3 requires two programs to create and maintain translations, ezlupdate and linguist. (You may not need ezlupdate.) These programs are based on the same tools from the Qt toolkit by Trolltech. The unix version of this toolkit is released under the GPL. eZ systems provides binaries for Windows and Mac OS X, see the translations page on ez.no/developer.

The linguist is unmodified from the Qt original, so you can also get it from other sources, such as RPMs. If you run Linux or a similar system, you will find the linguist as part of qt-3.*, qt-devel-3.* or a similarly named package.

The ezlupdate program is modified to make it understand eZ publish files. You will find the source code and build instructions in support/lupdate-ezpublish3 in the eZ publish distribution.

Making translations

Note: The database content of eZ publish is not translated, meaning that for instance class attributes are shown in English wherever this information is visible. This is normally only in the admin interface. The classes provided with eZ publish are merely examples provided to get you up and running quickly. You are encouraged to extend and/or replace these classes with your own classes. If you need a non-English administrative interface, you can translate your classes in the "Setup" section. (The translation system covers content in templates and PHP code.)

You must decide the locale code of your language. eZ publish 3 uses locale codes on the form aaa-AA, where the 3 first lowercase letters describe the language, while the last two uppercase letter describe the country in which the language is spoken. For instance, English as it is spoken in Great Britain would be eng-GB, while US English is eng-US.

Countries are specified by the ISO 3166 Country Code: <http://www.iso.ch/iso/en/prods-services/iso3166ma/index.html>

Language is specified by the ISO 639 Language Code: <http://www.w3.org/WAI/ER/IG/ert/iso639.htm>

You can also create a variation of a locale, you will for instance find two variations of nor-NO, nor-NO@intl and nor-NO@spraakraad, that are slight modifications of the original.

For the rest of this part, I assume you are translating nor-NO. (Just replace nor-NO with your locale.)

Copy share/locale/eng-GB.ini to share/locale/nor-NO.ini. Edit this file with a text editor. Here you set locale details such as time formats, currency and the names of the week days.

There are two methods to make your own .ts file. The fastest is to copy it from the share/translations/untranslated/translation.ts file which should be included in your distribution. Copy it to share/translations/nor-NO/translation.ts, and start translating it with the linguist, see further down.

The second method involves building the .ts file yourself, using ezlupdate. You need to do this if

you don't have the untranslated translation.ts file, or if you have changed template text using i18n operators, or if you use i18n in your own custom template or PHP code.

Enter the main eZ publish directory in a terminal and type

```
bin/linux/ezlupdate -v nor-NO
```

(Provided that nor-NO is your locale, of course. -v is for verbose, showing messages about what happens. You can also use -vv for extra verbose output, or omit it for silent behavior. Run bin/linux/ezlupdate -h for an explanation of the arguments.

You will now find a translation in share/translations/nor-NO/translation.ts or similar for other locales. Open this file in linguist and do the translation.

You will find documentation on linguist on Trolltechs page: <http://doc.trolltech.com/3.0/linguist-manual-3.html>

When you are done translating in linguist (or earlier, if you want to test part of a translation), open settings/site.ini. Go to the section [RegionalSettings] and set Locale=nor-NO. Also set TextTranslation=enabled, or the default (eng-GB) will be used.

Sample entry in settings/site.ini:

```
[RegionalSettings]
Locale=nor-NO
TextTranslation=enabled
```

If you run a multi-language site, you will also need to translate content objects. Set ContentObjectLocale=nor-NO if you want the default language to be nor-NO. Important: Before you do this, you should make sure that the new locale is added to the system. Go to Setup -> Translations and add your locale here if it does not exist. You should also translate the most used objects of your site before you change the ContentObjectLocale. To translate an object, edit it, and click "Edit" under "Translations" in the right-hand menu.

To distribute your translation, create a compressed archive, for instance .zip or tar.gz, of these two files:

```
share/locale/nor-NO.ini
share/translations/nor-NO/translation.ts
```

You could for instance do it like this:

```
tar -zcvf nor-NO.tar.gz \
  share/locale/nor-NO.ini share/translations/nor-NO/translation.ts
```

Installing translations

To install a translation, simply unpack the package and set the appropriate entries in settings/site.ini. These are the keys you need to change to enable the translation:

```
[RegionalSettings]
Locale=nor-NO
TextTranslation=enabled
TranslationCache=enabled
```

How to setup a Multilingual Site

EzPublish 3.1+ has made multilingual site implementation rather easy, though it is not straight forward or well documented. This document is an attempt to begin to systematize the knowledge and experience surrounding multilingual websites and offer a space whereby others can contribute to the community documentation. I am a relatively new ezp user and will try to make this as user-friendly as possible. This document represents a compilation of notes, documentation, forum messages and experiences.

I have broken down the aspects of multilingual websites into the following categories:

- methods for achieving a multilingual website
- administration of a MLW
- configuring and setup for a multilingual websites (MLW)

Methods for achieving a multilingual website

EzPublish 3 makes managing a MLW easy from the admin area. The tricky part seems to be in getting it set up correctly and then being able to display the content as desired. Based on what others have tried, there seem to be two general approaches to the former in ezpublish3. The first is the method that was built into ezp3 and primarily involves configuring ezpublish (through the siteaccess and site.ini settings) to display the appropriate language based on the url address. The other method involves the use of information gathered from the user navigating the site and depends on the use of modules not currently included with ezpublish. As I do not know of any public modules and want this to be as accessible as possible, I will focus on the former (siteaccess) method.

Administration of a MLW

Once the MLW is setup, administering a multilingual site (done through the admin area) involves two main tasks:

- establishing the languages for translations/languages for your site

While you are logged in to the admin area, go to Setup and click on the Translations button in the left column. Here you can add new languages and locale settings for your site (this may be unnecessary if you installed the multilanguage site as described below). To do so, you need to select a translation (or the 'custom' option), enter its name (ex. 'Spanish'), and locale information

(this is done using a code composed of a three letter identifier for the language and a two letter, uppercase, identifier for the country, such as eng-US (English, USA), esl-ES (Spanish, Spain) - see links below. Press 'Create' and you are done. Now a translation for that language can be added to any of your site's content.

Countries are specified by the [ISO 3166 Country Code](#)

Language is specified by the [ISO 639 Language Code](#)

managing and creating translations for your content

Now while editing your content, you can click on 'Manage' in the Translations box of the menu on the right hand side. There you can add a translation for the content object and translate it (with the original text beside what you are translating). **Don't forget to publish it!**

Translations are not versioned independently. If you want need that feature, it is recommended that you use different objects for different languages. You can then link the objects to their counterpart in other languages using the related objects function (also in right-hand menu). Also, this solution has advantages if your multilingual site has certain workflow demands. For example, having separate content objects for each language is also the best solution if you want translations made while the main language is published. In this case, the site manager would tell the translators to copy the original object and create the translations.

Please note that in version RC2 of eZpublish3, there was a bug where you would get no translations if the current language was placed before the others in the translation list.

Configuring a Multilingual website

When you first install eZpublish, the setup and installation tool lets you chose to make your site multilingual (and also which language should be the principle language for your site). If your translation does not ship with the eZpublish distribution, install it before installing eZpublish (see below).

Installing a translation

There are many translations for the eZpublish administration area available. Please see links below to find your translation and for instructions on how to easily install a translation. If you install the translation before installing eZpublish, the setup wizard will automatically configure the admin area to your preferred language and configure the site to allow all your other chosen translations for text. It is advisable, though not necessary, to setup the multilingual site at the installation because it is easier and helps avoid database errors that can be caused by changing language configurations.

If you install translations after your initial setup, you can still set up the site as multilingual. After you download and uncompress the translation files, you should set configurations according to your preferences (principal language, admin area language, etc). By setting your Locale variable, you can deploy the preferred language for the site admin area. By setting the ContentObjectLocale, you set the language for the content you develop. If you are unsure, do not change the later.

Where to get [translations](#)

How to [install a translation](#) or [here](#).

Siteaccess method setup

The site access settings let ezpublish receive configuration information based on the url address that is passed to it. So that ezpublish knows which language to present, we must configure the site access settings to present content in a certain language based on the url. The way this is done for multilingual sites is the same as for hosting several sites on one ezpublish installation (the difference being you do not necessarily have different content architecture in MLW). There are several ways this can be done in site access (see all of them at [here](#)).

I describe the process in three steps:

1. Tell ezpublish how to figure out which language to display (by the url used)
2. Create and edit siteaccess files and folders.
3. Fine tune

Step by step:

Step 1.

A typical, simple configuration (though not the only) for a multilingual site is to have each language have a distinct directory after the main address to access a certain language (ie, `site.com/index.php/en`, `site.com/index.php/es`, etc). To configure this setup, we start by editing `settings/override/site.ini.append` with (some of this may have been done for you during the installation, adjust as necessary):

```
[SiteAccessSettings]
#this part tells how the site access setting is chosen, in this case,
by the uri
MatchOrder=uri
URIMatchType=element
URIMatchElement=1
URIMatchRegexp=^/([^\s/]+)/
URIMatchRegexpItem=1
#this part tells ezp which siteaccess folders are available
AvailableSiteAccessList[]=en
AvailableSiteAccessList[]=es
AvailableSiteAccessList[]=br
AvailableSiteAccessList[]=admin
DebugAccess=enabled

[RegionalSettings]
#allow translations
TextTranslation=enabled
#cache the translations, can be disabled if desired (can take a long
time the first time the language is accessed)
TranslationCache=enabled

[ContentSettings]
#translations available for the content
TranslationList=eng-US;esl-ES;por-BR
#you can enable this after you have everything setup, leave it disabled
to see if you did it right!
ViewCaching=disabled
```

That is it! Some alternative setups are possible too (each of these change slightly the SiteAccessSettings configuration shown above). Again, see the site access documentation to figure out how:

- Virtual host (www.site.es, www.site.br or www.en.site.com, www.es.site.com, etc)
- Index (index_en.php, index.es.php)

Step 2.

Create one siteaccess folder for every language. The configuration files in these folders give eZpublish the correct language settings when the site is accessed. The folder names must be the same as the url directory name setup in step 1. In this example (site with English, Spanish and Portuguese), the resulting folders look like this:

```
settings/siteaccess/en   (English)
settings/siteaccess/es   (Spanish)
settings/siteaccess/br   (Portuguese)
```

(note*this may be in addition to your admin folder and configuration, which can also be configured with language information)

In each folder create a site.ini.append.php file. In the site.ini.append.php you can specify any number of values to differ from the main/default setup, such as different design folders, languages, login preferences. Take a look at settings/siteaccess/demo/site.ini.append and settings/siteaccess/admin/site.ini.append for examples. Whatever your other configuration decisions, the crucial part for multilingual is to set the Locale and ContentObjectLocale you want to use for that specific area of your site. For example, in my example case, the primary language is Spanish (for admin and user area, which are set in the settings/override/site.ini.append file). For the English part of my site, I edit settings/siteaccess/en/site.ini.php to have:

```
[RegionalSettings]
Locale=eng-US
HTTPLocale=en
ContentObjectLocale=eng-US
```

Next do the same for each respective siteaccess/whateverlanguage/site.ini.php file.

Text translation is automatically disabled when the Locale is set to eng-GB in settings/site.ini. Since we will need text translation, we have to change these variable settings in the settings/site.ini (otherwise the above work will not take effect). In settings/override/site.ini.append(.php) you can't override Locale, HTTPLocale and ContentObjectLocale; these are set in the siteaccess ini files. So first, in the settings/site.ini file, we comment out the following parts (with the '#' mark):

```
[RegionalSettings]
#Locale=eng-GB
#HTTPLocale=
#ContentObjectLocale=eng-GB
```

Step 3.

Fine tuning

If you are setting up several sites with one eZ publish 3 installation using site access, you might

want to separate the cache and storage directory. This will make things less cluttered and easy to manage when having many sites per installation. The default settings in eZ publish 3 stores cache and files in a common directory for all sites. This can cause problems if you e.g. want to move one of the sites to another server in the future. This can be changed by setting some variables in your site.ini.append for each of your sites.

It is important that both the user and the admin site has the same storage directory. Otherwise if you e.g. upload a new file using the admin site, the user site will not be able to locate the file because it is looking for it in the wrong directory. Also, make sure that Apache has sufficient rights for the folders you create.

For example - separate storage files and cache for 'site.com/es', 'site.com/en', 'site.com/br' and 'site.com/admin'.

Please see the [original of this](#)

Non latin languages and unicode:

If you need a language that does not use the latin character set, you will have to change this in settings/i18n.ini which defines the core charset for the site. With UTF-8 (Unicode) you can get most languages in the world on the same site at the same time.

See [Unicode with Ezpublish](#)

Cron jobs

Cron jobs in eZ publish are used to run workflows, notifications and other jobs which should be run periodically.

Running the cronjobs from shell

You can run cronjobs in eZ publish with PHP compiled as CGI. Example of cronjob execution (linux) is shown below.

```
/usr/bin/php -C runcronjobs.php
```

This is the most common place for the php executable. If you have troubles finding your PHP executable use:

```
whereis php
```

```
sample output: php: /usr/bin/php /etc/php.ini /usr/lib/php  
/usr/include/php /usr/share/php /usr/share/man/man1/php.1
```

You can configure which cronjobs should be executed in settings/cronjobs.ini. Cron job scripts are placed in cronjobs/ in the eZ publish root.

Starting the cronjob using HTTP (via Webbrowser)

If you do not have PHP compiled as a command-line-version, but only as a webserver module, you must run your cronjobs by calling the runcronjobs.php by a HTTP request from a webbrowser. You can do this manually or scheduled by a cronjob on the same or some other server, using lynx or wget or w3m to call the URL.

Note that this is **not secure**, as someone could call the URL very often and overload the server. You can combine the setup with a shell-driven call in order to be able to call the cronjobs, whenever you need them (for testing workflows, RSS imports etc etc).

So, you need to call "runcronjobs.php" by a HTTP request. In a "virtual host" ezPublish setup, this will not be possible, because the rewrite rule will always redirect you to "index.php". In order to re-gain access to "runcronjobs.php", you can change the existing rule to "not change request to *.php" or introduce an additional RewriteRule.

First option would look somehow like this:

```
RewriteRule !\.(php|gif|css|jpg|png) /path/to/ezpublish/index.php
```

This will allow execution of any PHP script and so, should not be used.

Better, but still not perfect, is to introduce a rewrite rule to call the runcronjobs.php:

```
RewriteRule ^/mySecretCronURL$ /path/to/ezpublish/runcronjobs.php [L]
RewriteRule !\.(gif|css|jpg|png)$ /path/to/ezpublish/index.php
```

The first line will look for a HTTP request to /mySecretCronURL and start the runcronjobs.php script; other requests are handled as usual (2nd line). This should do all the scheduled things and print some information about what it's doing to your webbrowser window. As this is no HTML, it will be ugly, so switch to "source code view" to see, what happened.

While this works, it's not secure, as everyone can call the URL, once it is know. I think, RewriteRules can be "conditional", which could secure that a bit by only using the RewriteRule for a know, static IP-Adress or so, but I'm not into this. Someone please add to this post, when he has more knowledge about it.

Finally, there is another method if you do not have cronjobs installed or CGI enabled php:

Automaticly run the cronjob using HTTP (via Webbrowser)

As long as your site is visited by your users, the index.php will always run each time a user

request something from your eZ site. Ok, then let's make index.php to automatically invoke the cronjobs, add these two line to the bottom of index.php, just before the line: eZExecution::cleanup();

```
include_once(
'kernel/classes/notification/eznotificationeventfilter.php' );
    eZNotificationEventFilter::process();
```

Using this method, you do not need to change the rewrite rule so it is as secure as before. But still there is the problem of serious performance impacts: every time your user request a page, the cronjobs will run.

This can be solved if we only run the cronjobs every, say 2 hours, so we will extend above two lines to followings:

```
/* Run Notification Filters */
$notifyFilterLastRunTimeFile = 'ezfilternotify_last_runtime';
$filterAutoRunInterval = 7200; //2 hours
$runFilter = false;
if( file_exists($notifyFilterLastRunTimeFile) )
{
    if( time() - filemtime( $notifyFilterLastRunTimeFile ) >
$filterAutoRunInterval )
    {
        $runFilter = true;
    }
}
else
    $runFilter = true;

if( $runFilter )
{
    include_once(
'kernel/classes/notification/eznotificationeventfilter.php' );
    touch($notifyFilterLastRunTimeFile);
    eZNotificationEventFilter::process();
}
```

That's it. As long as your site is visited every 2 hours, your cronjobs will be executed properly (of course, if no one visit your site, you can not expect anything coming out from these piece of code).

Login handler

After version 3.2, eZ publish supports login using external server such as LDAP or simply a text file which contains user account information. You can also make your own login handler.

Using LDAP server

In eZ publish configuration file *settings/site.ini*, find line

```
LoginHandler[]=standard
```

under [UserSettings] block.

To make your eZ publish support LDAP, add a new line under this line like following:

```
LoginHandler[]=standard  
LoginHandler[]=LDAP
```

This means that eZ publish will try to use 'standard' login handler as usual. If it does not succeed, it would connect to LDAP server to fetch user account information. You can also uncomment the first line like following since when you use LDAP server to login, eZ publish will try to find local user at first:

```
#LoginHandler[]=standard  
LoginHandler[]=LDAP
```

The next step is to configure LDAP settings in *settings/ldap.ini*. Here is an example:

```
[LDAPSettings]  
# Set to true if use LDAP server  
LDAPEnabled=true  
# LDAP host  
LDAPServer=  
# Port nr for LDAP, default is 389  
LDAPPort=389  
# Specifies the base DN for the directory.  
# Example: LDAPBaseDn=ou--sales,o--my company  
LDAPBaseDn=  
# Could be sub, one, base.  
LDAPSearchScope=sub  
# Use the equal sign to replace "=" when specify LDAPBaseDn or  
LDAPSearchFilters  
LDAPEqualSign=--  
# Add extra search requirement. Uncomment it if you don't need it.  
  
# Example: LDAPSearchFilters[]=objectClass--inetOrgPerson  
LDAPSearchFilters[]  
# LDAP attribute for login. Normally, uid  
LDAPLoginAttribute=uid  
# Could be id or name  
LDAPUserGroupType=id  
# Default place to store LDAP users. Could be content object id or  
group name for  
# LDAP user group, depends on LDAPUserGroupType.
```

```
LDAPUserGroup=12
# LDAP attribute type for user group. Could be name or id
LDAPUserGroupAttributeType=name
# LDAP attribute for user group. For example, employeetype. If
specified,
# LDAP users will be saved under the same group as in LDAP server.
LDAPUserGroupAttribute=employeetype
# LDAP attribute for First name. Normally, givenname
LDAPFirstNameAttribute=givenname
# LDAP attribute for Last name. Normally, sn
LDAPLastNameAttribute=sn
# LDAP attribute for email. Normally, mail
LDAPEmailAttribute=mail
```

Specify LDAP server, port number if not using 389, base dn, attribute matches between LDAP server and eZ publish user account, then LDAP login will be supported by eZ publish.

To manage users from LDAP server, a cron job will be executed periodically. You can configure *settings/cronjobs.ini* to disable running the script.

Note: LDAP support in PHP is not enabled by default. You will need to use the `--with-ldap[=DIR]` configuration option when compiling PHP to enable LDAP support. DIR is the LDAP base install directory.

Using text file

In eZ publish configuration file *settings/site.ini*, find line

```
LoginHandler[]=standard
```

under `[UserSettings]` block.

To make your eZ publish support text file, add a new line under this line like following:

```
LoginHandler[]=standard
LoginHandler[]=textfile
```

This means that eZ publish will try to use 'standard' login handler as usual. If it does not succeed, it would fetch user account information from a supplied text file. You can also uncomment the first line like following since by default, eZ publish will try to find local user at first:

```
#LoginHandler[]=standard
LoginHandler[]=textfile
```

The next step is to configure text file settings in *settings/textfile.ini*. Here is an example:

```
[TextFileSettings]
# Set to true if use the textfile login handler
TextFileEnabled=true
# textFile name
FileName=myusers.txt
# Where to find the text file, could be any directory under your eZ
publish
# root directory. If the FilePath is root, the file should be saved
under
# eZ publish root directory.
# Example: FilePath=var/storage
FilePath=root
# Could be tab, ';', ',', '\ ' or other string separator
#Example: FileFieldSeparator=;
FileFieldSeparator=tab
# Could be id or name
DefaultUserGroupType=id
# Default place to store users from textfile. Could be node id or group
name,
# depends on DefaultUserGroupType.
DefaultUserGroup=12
# Field column nr for login.
LoginAttribute=1
# Field column nr for password.
PasswordAttribute=3
# Field column nr for First name.
FirstNameAttribute=4
# Field column nr for Last name.
LastNameAttribute=5
# Field column nr for email
EmailAttribute=2
```

Specify file name, file path, file separator, and attribute matches between text file and eZ publish user account, then users saved in this text file could login into eZ publish.

How to make your own handler

Step 1: Create the necessary file for your new handler.

Suppose you are going to make a login handler called 'MyLoginHandler'. The one file you need to create should be *ezmyloginhandleruser.php* and it should be put under *./kernel/classes/datatypes/ezuser/*. The rule is that if you added a handler x, it should be stored as: *./kernel/classes/datatypes/ezuser/ezxuser.php*. Here is the standard structure of this file:

```
<?php
include_once( "kernel/classes/datatypes/ezuser/ezusersetting.php" );
include_once( "kernel/classes/datatypes/ezuser/ezuser.php" );
include_once( 'lib/ezutils/classes/ezini.php' );
```

```

class eZMyLoginHandlerUser extends eZUser
{
    /*!
     * Constructor
     */
    function eZMyLoginHandlerUser()
    {
    }

    /*!
     * \static
     */
    function &loginUser( $login, $password, $authenticationMatch = false
)
    {
        // Your implementation of the handler. If login allowed,
        // create the user and return $user object, return false if
login not
        // allowed.
        return $user;
    }
}

?>

```

Note that this class should extends eZUser and override function loginUser() for class eZUser.

In addition, you can create configuration file *settings/myloginhandler.ini*, which defines setting for your handler (user placement, attribute match between eZ publish and external server, etc.). When implement function loginUser(), you can then reading settings from this file.

Step 2: Update *settings/site.ini* to include the handler.

Open eZ publish configuration file *settings/site.ini*, find line

```
LoginHandler[]=standard
```

under [UserSettings] block.

Add 'LoginHandler[]=MyLoginHandler' under this line:

```
LoginHandler[]=standard
LoginHandler[]=MyLoginHandler
```

Search engine

Configuring the search engine.

Configuring binary file indexing

Note 1: Binaryfile indexing is available from 3.2.

Note 2: Before uploading any binary file, make sure, that the class attribute, that holds the file, is "searchable".

Note 3: As of 3.2 final, the filename itself is *not* indexed. But -for "file" objects- it will be found by the search engine, because usually the object name is taken from the filename.

The indexing engine currently supports *plaintext*, *PDF* and *MS Word* documents.

All settings for binaryfile indexing are found in the ini file *binaryfile.ini*.

plaintext

Plain text documents do not require any configuration as it will be used for all file that have the MIME-Type text/plain.

The default setting for plain text is

```
[HandlerSettings]
MetadataExtractor[text/plain]=plaintext
```

PDF

PDF files are handled using external programs which returns the content of the PDF file as plain text. This has been tested with the *pstotext* und *pdftotext* programs, but should work with others as well.

pstotext

The *pstotext* program can be found either on *freshmeat.net* or the *pstotext homepage*.

http://freshmeat.net/projects/pstotext/?topic_id=849

<http://research.compaq.com/SRC/virtualpaper/pstotext.html>

The default settings for using pstotext are

```
[HandlerSettings]
MetadataExtractor[application/pdf]=pdf
```

```
[PDFHandlerSettings]
TextExtractionTool=pstotext
```

pdftotext

Another option for indexing PDF files is pdftotext from the xpdf project. Read more about that [here](#).

MS Word

MS word documents are also handled using external programs. This feature requires the *wv* program to work properly.

WV

The *word view* program can be found either on *freshmeat.net* or the *wv homepage*.
http://freshmeat.net/projects/wv/?topic_id=849
<http://wvWare.sourceforge.net/>

The default settings for using *wv* are

```
[HandlerSettings]
MetaDataExtractor[application/msword]=word

[WordHandlerSettings]
TextExtractionTool=wvWare -x /usr/local/wv/wvText.xml
```

Alternatives:

You may consider these alternative indexers:

For MS Word: antiword (<http://www.winfield.demon.nl/> or <http://www.antiword.org>)

For MS Excel: XLhtml (<http://chicago.sourceforge.net/xlhtml/>)

For MS PowerPoint: XLhtml (<http://chicago.sourceforge.net/xlhtml/>), which also includes a PPT->HTML converter

For XLhtml, you will need a html->Text-Converter. You can use a webbrowser like lynx or w3m for this.

As soon as we have set this up completely (on ez 3.2) and integrated everything with ez3.2, i will give more detailed info.

Issues:

For problems with large files, see

http://ez.no/developer/ez_publish_3/bug_reports/weird_search_limitations_binary_file

Note, that indexing relies on correct MIME-Types. As of 3.2-3, the MIME-Type for MS Excel-Files was not set.

Solving problems with binary file indexing

When binary file indexing is not working, there are a few things to check.

Large files are not indexed or do not turn up in the search results	MySQL packet size too small	Increase the packet size (1M by default) to a larger value (16M is the maximum allowed)
Acrobat pdf files are not properly indexed	ghostscript (older version) is used	Install the xpdf package and use pdftotext in a custom script.
Acrobat pdf files take ages before they are indexed	ghostscript is used	See solution above
Sometimes, a file (pdf, word, ...) is not indexed, with other users there are no problems	The mime settings of the browser may be wrong	Edit the browser settings (look for spurious mime types connected to a certain filename extension for mozilla based browsers)
MS Excel files are not indexed (ezP 3.2)	MIME-Type for Excel is not defined in 3.2	Open <code>\\lib\ezutils\classes\ezmimetype.php</code> and insert <code>array('xls', 'application/vnd.ms-excel'),</code> into the Array <code>\$QuickMIMETypes</code> at the end of the file. Note: Only newly uploaded files get the MIME-Type set, so this will not help you for existing files.

Delayed Indexing

only available in 3.4 or higher

When the setting "DelayedIndexing" is set to "enabled", eZ publish will no longer index your document if you are publishing it, but instead will put it in a queue for indexing this document on a later time. This will reduce the time editors have to wait when publishing a document, as the time to index one costs quite some time for larger documents.

In order to start this process you will need to enable the cron job "indexcontent" by adding the following to `settings/override/cronjob.ini` in the section "CronjobSettings":

```
Scripts[]=indexcontent.php
```

For setting up the cronjob, please refer to:

http://ez.no/ez_publish/documentation/installation/the_cronjob_script

Wildcard search

In eZ publish 3.4 wildcard search is supported, but not enabled by default.

To enable wildcard search, override *site.ini* to :

```
[SearchSettings]
EnableWildcard=true
```

No reindexing is needed.

This will enable usage of the wildcard character *. The search system only support placing the wildcard character at either start or end of words.

After wildcard search has been enabled, it's possible to search for terms like `pub*`

`pub*` will return all matches for all words starting with `pub`, ex: `pub`, `pubs`, `puben`, etc.

Searching for `*pub` will return all matches for all words ending with `pub`, ex: `testpub`, `pub`, etc.

Due to server load a minimum of 3 regular characters must be in the search term using wildcards. This can be changed by overriding **site.ini**

```
[SearchSettings]
MinCharacterWildcard=3
```

Known issue:

When searching for two or more words containing wild cards, all objects containing the wildcards will be returned.

Tips & Tricks

A collection of common configuration tricks you can use.

Manual "modfix"

If I cannot use modfix.sh how do I have to set permissions / create cache folders?

If the following directories do not exist, please create them e.g. with your ftp program.

```
var/cache/ini
var/cache/texttoimage
var/cache/translation
var/cache/template/tree
var/cache/template/process
var/log
```

Please set the permission of the following directories to 777, e.g. with you ftp program.

```
var/cache/texttoimage
var/cache/translation
var/cache/ini
var/log
```

Set the permissions of the following directories including ALL subdirectories to 777.

```
var/storage
settings
```

Set the permissions of the following files to 666.

```
var/log/error.log
var/log/warning.log
var/log/notice.log
var/log/debug.log
```

Several sites with one installation

If you want subdomains (sub.domain.tld, sub2.domain.tld, ...) to work as separate sites that run on one installation of eZ publish 3.x you do the following:

Edit settings/site.ini.php

```
[SiteAccessSettings]
MatchOrder=host
HostMatchRegexp=^(.+\\\.+)\\.domain\\.tld$
HostMatchSubtextPost=\\.domain\\.tld
```

If you would like to use domains (www.domain1.tld, www.domain2.tld) to work as separate sites that run on one installation of eZ publish 3.x you do the following:

Edit settings/site.ini.php

```
[SiteAccessSettings]
MatchOrder=host
HostMatchType=map
HostMatchMapItems[]=domain1.net;www
HostMatchMapItems[]=www.domain1.net;www
HostMatchMapItems[]=admin.domain1.net;admin
HostMatchMapItems[]=www.domain2.com;domain2
```

As you can see this method allows for subdomains and domains (and you don't have to worry about learning regexp).

Then you just create the siteaccess files for each site you want to host. E.g.

```
settings/siteaccess/sub/site.ini.append
settings/siteaccess/sub2/site.ini.append
```

In the site.ini.append you can specify any values to differ from the main/default setup, even a completely different database or different design folders. Take a look at *settings/siteaccess/demo/site.ini.append* and *settings/siteaccess/admin/site.ini.append* for examples.

Enable / disable the setup wizard

How do I disable/enable the setup function? How do I rerun the setup wizard?

In settings/site.ini.php find the following:

```
[SiteAccessSettings]
CheckValidity=false
```

If the value is set to false, the setup function is disabled, if set to true, it's enabled. Or in other words, if you want to rerun the setup modify the above value to true.

Note: eZ publish will create override files in settings/override when running the setup wizard. Make sure that you change values in these files.

Determine if user has to log in

How do I determine whether the user has to log in to view the site?

In settings/site.ini.php find the following:

```
[SiteAccessSettings]  
RequireUserLogin=true
```

This is the default setting. To change the value on the per site basis (probably more appropriate), go to *settings/siteaccess/(yoursite)/site.ini.append* and define the above value in this file to your liking.

Change the name of the main script

To change the name of the index.php to something else, e.g. default.php, you can do the following

1. Rename the index.php to default.php :-)
2. Edit the renamed script, in line 259 change to

```
eZSys::init( 'default.php' );
```

Thanks to Kai Dübber

Setting default page

How do I set the default view when URI matching was not successful?

In site.ini, under [siteSettings]:

```
# Which page to show when the root index is accessed
IndexPath=/content/view/sitemap/2/

# What to do when a module does not exists, use either defaultpage or
displayerror
ErrorHandler=displayerror
DefaultPage=/content/view/sitemap/2/

# Default access is needed when uri type matching is done, this is
# because with empty urls it's not possible to fetch the access
DefaultAccess=
```

So, you can set DefaultPage and DefaultAccess to match your needs.

Setting the path to the var directory

For some installations it might be appropriate to set the path to the /var directory. You can even set different paths for each site of one installation. Use the following directives in site.ini(.php/.append)

```
[FileSettings]
StorageDir=path/to/storage
CacheDir=path/to/cache
```

File permissions

To ensure your server install of eZ publish is on the road to becoming secure you can run the following commands that start to lock down your eZ publish 3 project directory.

I think to start with I need to pass on a saying that a colleague passed to me when dealing with security, it has stood the test of time for me.

If in doubt, don't!

This always come to mind before implementing a new feature or install a new package. If I don't understand it fully then I don't implement it on a production environment.

Before you run these commands ensure you are in the directory above you eZ publish 3 project directory. do NOT run them in any other location

1. Change ownership

```
chown -R apache ezpublish_project
```

Where ezpublish_project will be the directory your project is contained

```
chgrp -R mygroup ezpublish_project
```

Where mygroup is the group you belong to, this means you can do updates without login in a root which is a big no no.

2. Change access rights

```
chmod -R 570 ezpublish_project
```

This ensures apache has read but not write access to your main project

```
chmod -R 770 ezpublish_project/var
```

This ensures apache can create those all important images and life saving cache files.

3. Get reading

You need to read and understand what you have just done. I recommend treeware from O'Reilly for GNU/Linux admin, you can pick up from your local bookstore or online. If you are on GNU/Linux by running the following commands that bring up manual pages or online at <http://www.tldp.org/docs.html#howto>

```
man chown
```

```
man chgrp
```

```
man chmod
```

As you can imagine this is only the beginning, if you run your own server then you need to lock down MySQL/postgreSQL, PHP, Apache, SSH and of course the other services you are running plus fix/patch updates.

Defining design resources

Under settings/siteaccess/(site access) you find a file called site.ini.append. For each site you set up with this installation of eZ publish, you can define the design folder to go with it.

Follow this link to [clarify the difference](#) between "site design" and "site access"

For site access "user" you find under **/settings/siteaccess/user/site.ini.append** the following section

```
[DesignSettings]
SiteDesign=user
```

Change the **user** above (Site design) to any folder name you like to point to your new design. Of course you need also a design folder for this like: design/(your site design)/. And inside this folder you need a template folder with the pagelayout.tpl in it. Take a look at [Custom designs - directory structure explained \(3.x\)](#).

To change the default folder for the default site, go to /settings/override/site.ini.append.php and add or edit

```
[DesignSettings]
StandardDesign=standard
SiteDesign=standard
```

to change to the appropriate folders.

Exclude URIs from rewrite

If you have set up virtual hosts and want to exclude certain URIs from the rewrite, e.g. for other scripts like a webmail script at <http://www.example.com/webmail> or a stats script at <http://www.example.com/stats> you can do the following.

In your httpd.conf rewrite rule use

```
RewriteRule
!(^/stats/.*$|^/webmail/.*$|^(gif|css|jpg|png|js|asf|avi|wmv|swf|xsl|jar))$ /<your path to>/index.php
```

for the above example URIs to be excluded from the rewrite.

You will need to avoid the words stats and webmail as first level nodes in your site when using url_alias (nice urls).

Thanks to Paul Borgermans

Security

After eZ publish is installed there are several things you can do to secure your site. Here you will find information about how to do so.

Non-Virtualhost

Securing the site depends on whether you are using a virtualhost setup or a non-virtualhost setup. A virtualhost setup means that all urls are redirected to the index.php script while non-virtualhost requires that the index.php script is mentioned in the url.

Virtualhost setups

Virtualhost setups are secure by default since all request are sent to index.php script, with the exception of images, stylesheets and javascripts. The only thing that is required for securing the site is to make sure all custom made templates follow the [security guidelines for templates.](#)

Non-Virtualhost setups

Non-Virtualhost setups are insecure by default. The index.php must explicitly be placed in the url, this means that any other scripts may be executed directly as well as open up .ini files with password information. Because of this it is not recommended to use non-virtualhost setups, however if you don't have any choice there are some guidelines which can be used to secure the site.

You also need to make sure that all custom made templates follow the [security guidelines for templates.](#)

Install a .htaccess file

The Apache webserver allows each site to install a .htaccess file which can control which files are accessible as well as set PHP options. The .htaccess file is placed in the root of your eZ publish installation, an example of how it may look follows.

```

<FilesMatch ". ">
order allow,deny
deny from all
</FilesMatch>

<FilesMatch "(index\.php|\.(gif|jpe?g|png|css|jar|js|html))$" >
order allow,deny
allow from all
</FilesMatch>

RewriteEngine On
RewriteRule !\.(gif|css|jpe?g|png|css|jar|js|html)$ index.php

DirectoryIndex index.php

```

Use .ini.php files

All .ini files in eZ publish are readable when in non-virtualhost mode, this means that placing items such as usernames and passwords in these files are dangerous.

Fortunately the .ini file reader in eZ publish supports reading so called PHP wrapped .ini files. This means to create a file with the suffix .ini.php (.ini.append.php for append files), wrap it in a PHP comment and place it in the settings (settings/override for append files) directory. For instance the web setup will automatically create such files in settings/override/ for you with all the personal settings.

site.ini.php

```

<?php /*
[DatabaseSettings]
Server=mydbserver
User=myuser
Password=mypassword
*/ ?>

```

Alternative: Non-Virtualhost URL without index.php

If your Server can run mod_rewrite, you can search ezurl and replace with ezroot in all your templates, put the .htaccess file below in your space then you'll have full nice url fun in nvH environment

.htaccess

```

DirectoryIndex index.php

<FilesMatch
"(index\.php|\.(gif|html|css|jpe?g|png|ico|js|asf|avi|wmv|swf|xsl|jar|pdf|doc))$" >
order allow,deny
allow from all

```

```

Options Indexes FollowSymLinks Includes ExecCGI
</FilesMatch>

RewriteEngine on
# we are reached via /<path-to-ezpublish>/ prefix
#RewriteBase /<path-to-ezPublish>/ #if ezPublish is not in root
directory take this and not the following
RewriteBase /

# IF URI does not end with "/" #####
RewriteCond %{REQUEST_URI} !/$
# and IF URI does not end with <anything> <dot> <something>
RewriteCond %{REQUEST_URI}
![^.*]*\.[php|html|gif|css|jpe?g|png|ico|js|asf|avi|wmv|swf|xsl|jar|pdf|d
oc|]+$

#RewriteCond %{REQUEST_URI} [index\.php]+$
# THEN append trailing slash and redirect the client
RewriteCond %{HTTP_HOST} (.+)
RewriteRule (.*?) http://%1/$1/ [L]

# first we rewrite the root dir to the handling php script
RewriteRule ^$ index.php [L]
RewriteRule ^index\.html$ index.php [L]

# strip out the subdirs when the browser requests us from per dir pages
#RewriteRule ^.+/<path-to-ezPublish>+/. $ $1 [L] #if ezPublish is not in
root directory take this and not the following
RewriteRule ^.+/. $ $1 [L]

# and now break the rewriting for local files
#RewriteRule ^<path-to-ezPublish>\.php.* - [L] #if ezPublish is not in
root directory take this and not RewriteRule ^\.php.* - [L]

# exclude here directories or files eg. your webmail, phpadsnew,
pphlogger
#start for 3.4, for 3.0 - 3.3 replace this part with code below.
RewriteRule ^\.php.* - [L]
RewriteRule ^design\..* - [L]
RewriteRule ^var/./storage\..* - [L]
RewriteRule ^var/storage\..* - [L]
RewriteRule ^var/./cache\..* - [L]
RewriteRule ^var/cache\..* - [L]
RewriteRule ^extension/./design\..* - [L]
RewriteRule ^kernel/setup/packages\..* - [L]
RewriteRule ^packages\..* - [L]
RewriteRule ^share/icons\..* - [L]
#end for 3.4

# anything else is a subdir which gets handled by another php script
RewriteRule !^index\.php.* - [C]
RewriteRule (.*?) index.php/$1

```

the 3.0 - 3.3 part:

```

#start for 3.0 - 3.3
RewriteRule ^\.php.* - [L]
RewriteRule ^design\..* - [L]
RewriteRule ^var/./storage\..* - [L]
RewriteRule ^var/storage\..* - [L]
RewriteRule ^var/./cache\..* - [L]

```

```
RewriteRule ^var/cache\.* - [L]
RewriteRule ^extension/*/design\.* - [L]
RewriteRule ^kernel/setup/packages\.* - [L]
#end for 3.0 - 3.3
```

More information and discussion on this part in forum:

[.htaccess - hidden at last](#)

Discussions

Security issues discussed in the forums at ez.no.

Two forum threads on recent security concerns [here](#) and [here](#)

Restrictions

How to disable certain functions on your site.

Require user login

If you would like to require the users to login before they get access to the site at all you can set the following in site.ini

```
[SiteAccessSettings]
RequireUserLogin=true
```

Allowing access

You can, and in many cases should, enable access to certain functions in eZ publish to all users. This is done with the PolicyOmitList. Here is an example configuration for this.

```
[RoleSettings]
PolicyOmitList[]=user/login
PolicyOmitList[]=user/logout
PolicyOmitList[]=user/forgotpassword
PolicyOmitList[]=user/register
PolicyOmitList[]=user/activate
PolicyOmitList[]=mymodule/myfunction
```

Disabling specific functions

You can disable certain functions site wide to make your setup more secure. You can e.g. disable user registering for your website.

```
[SiteAccessRules]
Rules[]
# Enable all functions
Rules[]=Access;enable
Rules[]=ModuleAll;true
# Disable setup module and user register
Rules[]=Access:disable
Rules[]=Module;setup
Rules[]=Module;user/register
```

System security

eZ publish systems can only be as secure as the hosting environment it lives on. It is wise to ensure that PHP, MySQL, Apache, PostgreSQL, IIS are as secure as they can be.

There are books and web sites on this subject, so if you run your own server it's wise to find out about these issues before you go into production.

Optimization

Here you will find information about how you can tune eZ publish to give the best possible performance in your setup.

PHP Acceleration

PHP Acceleration is caching of the compiled PHP script in memory. This is **highly** recommended when running eZ publish it will really boost the performance of your eZ publish site.

Collection of documents touching the optimization of eZ publish 3.

Execution speed increases a lot if you use the free [Alternativ PHP Cache](#), [ionCube.php accelerator](#) or [Turck MMCache](#).

IonCube can improve response times from the server by up to 80%, but make sure you check

your php.ini as it requires more memory and if your memory limit is too low this will cause script errors.

A good general setting for ionCude php accelerator is (in php.ini).

```
zend_extension=/usr/lib/php4/php_accelerator_1.3.3r2.so
phpa.shm_ttl = 6h
phpa.shm_size = 15
```

Its probably worth checking you /tmp directory to ensure you have enough space for all those files the accelerator creates.

If your memory gets out of control as it does on standard Redhat 7.3 installs then you need the memory workaround

```
phpa.enable_php_memory_bug_workaround = 1
```

ionCube is a good solid "free" solution when speeding up php and has been run by some community members for months without problem. It has been known to have problems on PHP 4.1.2 when you host many sites, ensure you test it fully before deploying 10 sites+ on a box with this accelerator.

An even better but expensive option (490+ USD) is the [Zend Performance Suite](#)

Other options include compiling PHP to remove unwanted modules, this will reduce the size of the executed code.

You might also consider [building your own Apache](#) for the use in combination of eZ publish.

Disabling the cache

If you do not have the *site.ini* files in *settings/override* you should create *settings/override/site.ini.append* and insert the following:

```
[ContentSettings]
ViewCaching=disabled
```

You may want also disable template cache

```
[TemplateSettings]
TemplateCache=disabled
```

and override cache

```
[OverrideSettings]
```

Cache=disabled

Alternative way to disable view cache

Remove the cache files in *var/cache* and change the permission of the *var/cache* folder to **000** with:

```
chmod 000 var/cache
```

Configuration tuning

In eZ publish you can configure quite a lot and you can tweak your site's performance by simple .ini file editing.

Charset conversion

eZ publish will automatically convert charsets between the different parts of the system. Database, templates, ini files etc are automatically converted if different charsets are used. However this conversion takes time. If you need to have charset conversion for some reason you should compile PHP with the mbstring extension, this will go a lot faster than the built-in conversion done in eZ publish.

If you don't need to do charset conversion you should make sure that eZ publish uses the same charset all over. The following settings/files should be the same.

```
site.ini:  
[DatabaseSettings]  
Charset=iso-8859-15
```

```
l18n.ini:  
[CharacterSettings]  
Charset=iso-8859-15
```

```
template.ini:  
[CharsetSettings]  
DefaultTemplateCharset=iso-8859-15
```

Disable workflow

Read workflows cause an overhead on your site. This can be disabled by removing read cache

from the workflow.ini file.

If you remove the content_read workflow from OperationSettings you will get an performance improvement.

Recommended setting, if you don't need content read workflows.

```
[OperationSettings]
AvailableOperations=content_publish;shop_confirmorder;shop_checkout
```

Caching for improved speed

eZ publish has support for cache on different levels. Template code, template result and views can be cached.

Template caching

There are two different caching mechanisms in eZ publish. Node tree caching and template compiling. These should not be used at the same time though.

Using eZ publish 3.1 or newer you should use template compiling, since this is the most effective way of caching template code.

You can enable caching in site.ini. Recommended settings from 3.1:

```
[TemplateSettings]
NodeTreeCaching=disabled
TemplateCompile=enabled
ProcessCaching=enabled
TemplateCache=enabled
```

Content cache

The content cache is result caching of node view. E.g. an whole article, forum message or product view page.

You can enable this in site.ini:

```
[ContentSettings]
ViewCaching=enabled
```

Speeding up acrobat pdf document indexing

When you make binary files searchable in release 3.2, you may notice a severe performance degradation when indexing pdf files. It will also take a lot of time on larger documents before your browser even returns when publishing an object which a searchable binary file attribute.

This is mainly due to the default use of pstotext which is based on ghostscript. For lots of pdf documents, the output is mainly garbage which is hard to index by eZ publish.

It is much better to use **pdftotext** from the **xpdf** project. However, to use it with eZ publish, you need to keep in mind the command line arguments needed.

Normally pdftotext expects an input filename and an output filename. However, the eZ publish implementation expects the output on **stdout** which is specified with pdftotext with a -

To make pdftotext work with ez publish, you need to create a small script, say **ezpdftotext** and put it in a place where the webserver/php can find it.

The contents of this file should be

```
#!/bin/sh
#ezpdftotext script
pdftotext $1 -
```

(You may need to specify the full path of "pdftotext" in the last line of the shell script above)

You can download xpdf with pdftotext from

<http://www.foolabs.com/xpdf/download.html>

if you do not have it already of course.

Note. Redhat 7.3 users will need to download source and compile as the shipped version of pdftotext does not convert PDF files very well.

[Tested on Openoffice 1.1rc4 output Screen optimised PDF files]

Your binaryfile.ini.append in the settings/override directory should include the following statements

```
[HandlerSettings]
#....
MetaDataExtractor[application/pdf]=pdf
#....
# The path to the text extraction tool to use to
# fetch the information in PDF files
[PDFHandlerSettings]
TextExtractionTool=ezpdftotext
```

Voila, happy pdf indexing (after clearing the ini cache)!

Backup & Restore

Information about backup and restore will come.

Move to new provider

This is just a list with things to do/remember if you want to move to another server, operating system or provider.

Do a fresh install

Start by doing a fresh install on the new server. Check if the site runs as expected and move on to the next step.

Move the databaseMySQL

The MySQL site has some information on **creating** a dump file from the root:[Dumping Table Structure and Data](#)

```
shell> mysqldump --opt EzPublish_Database > dump_file.sql
```

or if you want to just backup your data while you cms is running try the following, although its better if you want to restore quickly, it does remove everything that existed before.

```
mysqldump -uxxx -pyyy --add-drop-table --lock-tables --complete-insert  
mydata >/opt/backup/mydata.sql
```

If username, password or host are required, add `--host=host --user=user_name --password` between mysqldump and the database name.

How to **install** the dump in the new database:

```
shell> mysql database < dump_file.sql
```

If your providers allows remote MySQL access, ie. not just to localhost, try to **transfer** the dump to the new server:

```
shell> mysqldump --host=old_host --opt old_database | mysql --  
host=new_remote-host -C new_database
```

If this is too much for you or you're used to a GUI, try to use SQL managers like [phpMyAdmin](#) or [Webmin](#). Some providers have them installed by default, check their help for more details.

Note: when you **import** the dump files on the new database and you receive a timeout (default: 30sec), the dump file contains too many instructions for SQL to handle within the timeout. You need to split the dump file in smaller parts above the comments '#'.

PostgreSQL

If someone knows about PostgreSQL, please post here.

Take a look at [PostgreSQL](#)

Move the files

By default, your files are located in **./settings/**, **./design/{your design name}** and in **./var/storage**. Compress, with [WinZip](#) or a similar program, the files with *all* subdirectories. Move them from your site to the other provider with FTP. Don't forget the SQL dump file and to decompress the files.

Change the settings

Check in **./settings** all **.ini.php** files for the correct settings for the new provider. More information on the settings: [LINK ME - ezwiki](#)

If you experience access problems, rerun **./bin/modfix.sh** and check all file and directory settings.

Overview

Just an overview....

For a backup of an ez3 (or ez2) system, you can use the backup-script "[reoback](#)".

Backup(for ez2, optional:) Call your cron-job to do a session cleanup and so on, thus minimizing your session* tables in the DB
Do a backup of your Database. For MySQL, we use mysqldump to dump one ez Database into a text file.
Use reoback to backup your complete ez3 directory, the database dump, that was created by mysqldump and maybe your Apache configuration. This should give you one nice tar.gz of your site.

Restore

Reverse the backup process :

Unpack your tar.gz to re-create the files

re-recreate your database by importing your DB dump (like mysql my_database if necessary, restore your apache config (httpd.conf) and restart your webserver

General Notes:

For consistence's sake, try to backup the files at the same time or directly after the database.

Reoback can be used to copy or move your backup files to some other server using FTP or NFS.

I would recommend this. If you don't have some other server/webpace, that you can FTP to, you need to download the backup files to some client (probably manually, except your client is running all the time). Beware of huge files!

Reoback can be used to do incremental backups.

This process can be used to port/copy/clone any ez site. Just restore to some other place/server and adapt your .ini's.

Of course, you should automate the process by using the server's cron system and some small Perl scripts.

PS: Can any MySQL guru please comment, if mysqldump really is the best choice for doing a consistant dump (=is the DB locked)?**Regarding mysqldump:** If you use the -l or --lock-tables option when using mysqldump, it will lock all tables in a database (not across all databases, just one at a time). There's also the --add-locks option which will lock each table separately but it does not guarantee consistency across a database.

if necessary, restore your apache config (httpd.conf) and restart your webserver

General Notes:

For consistence's sake, try to backup the files at the same time or directly after the database.

Reoback can be used to copy or move your backup files to some other server using FTP or NFS. I would recommend this. If you don't have some other server/webpace, that you can FTP to, you need to download the backup files to some client (probably manually, except your client is running all the time). Beware of huge files!

Reoback can be used to do incremental backups.

This process can be used to port/copy/clone any ez site. Just restore to some other place/server and adapt your .ini's.

Of course, you should automate the process by using the server's cron system and some small Perl scripts.

PS:

Can any MySQL guru please comment, if mysqldump really is the best choice for doing a consistant dump (=is the DB locked)?**Regarding mysqldump:** If you use the -l or --lock-tables option when using mysqldump, it will lock all tables in a database (not across all databases, just one at a time). There's also the --add-locks option which will lock each table separately but it does not guarantee consistency across a database.

if necessary, restore your apache config (httpd.conf) and restart your webserver

General Notes:

For consistence's sake, try to backup the files at the same time or directly after the database.

Reoback can be used to copy or move your backup files to some other server using FTP or NFS.

I would recommend this.

If you don't have some other server/webpace, that you can FTP to, you need to download the backup files to some client (probably manually, except your client is running all the time). Beware of huge files!

Reoback can be used to do incremental backups.

This process can be used to port/copy/clone any ez site. Just restore to some other place/server and adapt your .ini's.

Of course, you should automate the process by using the server's cron system and some small Perl scripts.

PS: Can any MySQL guru please comment, if mysqldump really is the best choice for doing a consistant dump (=is the DB locked)?

Regarding mysqldump: If you use the `-l` or `--lock-tables` option when using mysqldump, it will lock all tables in a database (not across all databases, just one at a time). There's also the `--add-locks` option which will lock each table separately but it does not guarantee consistency across a database.

Troubleshooting

Here you will find typical problems and solutions for these.

404 - Page not found

If you are getting this after initial install and while trying to login for the first time or at some other point when you are starting to use eZ publish the error can mean one of three things.

a) If you are running Apache 2, then you must have `PATH_INFO` turned on explicitly.

See [eZ publish and Apache 2](#) and [this forum thread](#).

b) You are running PHP as CGI. There are problems related with running PHP as CGI and eZ publish currently (as of this writing) does not support this. Try searching the forums for CGI and you will get some hints as to how to get it running anyway. *Update: In the forum there is a [patch for version 3.02](#).*

c) If your PHP memory limit is not set high enough you might get the error as well. See [Not enough PHP memory](#).

Change the right configuration file

As described in [Configuration files and directories explained](#), you have to make any changes to your configuration in the correct ini files. `/settings/override` override `/settings/siteaccess` which override the standard files `/settings`.

So usually, you would **not** want to change anything in the standard config files in `/settings` but rather in `/settings/siteaccess` (per logical site) and in `/settings/override` (for overall settings).

Debug popup window infinite loops

If you setup a VHost eZ publish 3.1 using the apache vhost configuration modified from the samples in doc/INSTALL, enable DebugOutput in section [DebugSettings] and specify the popup debug mode, then the "eZ debug" window will infinitely load itself, and you just can see nothing useful.

Solution:
change the line like this:

```
RewriteRule !\.(gif|css|jpg|png|jar|js)$  
/your/ez/path/ezpublish/index.php
```

to

```
RewriteRule !\.(gif|css|jpg|png|jar|js|html)$ /index.php
```

This just means to disable rewrite when clients request a *.html file.

Since eZ do not have any other *.html files, so there should be no potential security leaks.

eZ publish 3 and Apache 2

There have been problems with Apache 2. The securest thing to do is to use Apache 1.3 instead. If you absolutely need Apache 2, try one of the following:

a) PATH_INFO must be turned on **explicitly** in Apache 2.*.

Two helpful URL's:

<http://httpd.apache.org/docs-2.0/mod/core.html#acceptpathinfo>

<http://www.apacheweek.com/issues/02-07-12>

b) Some versions worked, some didn't. Try switching to a different version.

According to Apache's bug database, this bug was fixed in 2.0.36.

However, I'm not completely convinced it is totally functional.

But good news, if adding 'AcceptPathInfo On' in the main apache config file fails to work then try putting the directive in .htaccess for the root of ezpublish.

Getting looped back when logging in

If you are getting looped back after logging in to the admin area, you are probably running PHP as CGI. Currently there are problems with this. Please run PHP as a web server module.

Discussed e.g. in [this thread](#). Note: there is a quasi-patch for version 3.02 in the forums to get eZ publish working with CGI (see [this thread](#) and [this user doc](#)).

Giving you back the admin rights

Maybe not so typical but critical nonetheless :-). Do a

```
insert into ezuser_role ( role_id, contentobject_id ) values(2,12);
```

It will assign the admin role to all users under the administrator user group. To assign the admin role directly to the admin user:

```
insert into ezuser_role ( role_id, contentobject_id ) values(2,14);
```

I need to reindex my site for search.

If your index is messed up and you need to reindex you first remove the old search entries via the sql commands.

```
delete from ezsearch_word;  
delete from ezsearch_object_word_link;
```

Then you need to re-index the search-index in eZ publish.

```
php -C update/common/scripts/updatesearchindex.php
```

Thats it, index is updated.

NOTE:

If you're running a multilanguage site or if the updatescript doesn't seem to index stuff: make sure that the script uses the settings for the correct siteaccess. Use the "-s" or "--siteaccess" parameter to set this. Use "--help" to see all available parameters, etc.

Images do not work

You've gotten eZ publish successfully installed, but you cannot see any images. Here are some hints to what you should check.

File permissions

eZ publish needs to create files when you upload images and when images are converted. This means that the apache (webserver) user needs to have write permissions to the directory **var/storage/** and it's sub directories.

Image conversion program

eZ publish can convert images with one of the two programs GD and ImageMagick. Check that these are correctly installed and working. Check your image settings in the configuration file **settings/image.ini**. (*Note that if there is an image.ini.append/.php in /settings/override you have to make changes there instead of the main image.ini!*)

[If your ImageMagick was detected by the setup guide, but still won't work, check the Sub-Document.]

Safemode and open_basedir

eZ publish does not work properly under safemode and/or with open_basedir restrictions. These needs to be turned off.

Image classes not available to eZ

If you only see a Link and not the image it could be that you do not have the image display classes.

Copy **embed_class_5.tpl** and **text_linked.tp'** from **/design/admin/override/templates/content/view** to **override/templates/content/view/** in Example:

```
cp /ez/design/admin/override/templates/content/view/*
```

Uploading new images

If you have no problems with images from demo design, but you can't upload new images or change existing images in demo design, check first settings in **php.ini** file. In the part *File Uploads* ought to be:

file_uploads = On

For more informations about php.ini settings see [here](#).

Incorrect image size for conversion

If you have images in your original and reference directories then the problem could be you image sizes. Ensure you have sizes for both height and width in your site.ini.php file. Without both sizes image convert will fail.

Image Scaling (incl. "ScaleLargerThanOriginal")

There is a setting called "ScaleLargeThanOriginal" (formerly: "ScaleLargerThanOriginal"), which you can set to "false" in order to limit an image variation size to the original size (so the variation will never be larger than the original image).

This setting is only used with ImageMagick's "convert". It is ignored, when you use the PHP-built-in GD for image handling.

Note: When "ScaleLargerThanOriginal" is set to "false", then command line used to call "convert" includes a ">" character (which tells convert "don't enlarge").

For example, the call to convert looks like this:

```
home/usr/bin/convert -geometry "150x150>" inputfile outputfile
```

As the > character usually means an output redirection on OS-level, it may cause trouble (read: error) in some situations. Particularly, this will fail, when safe mode is on. But there are other situations known (and not yet solved), where this causes an error, despite safe_mode being off. An error here means, that ezPublish cannot create a reference image and so, no image variation can be created. The original image file is uploaded nicely, but you will not see anything on your website, except, when you use the image-size "original".

GD fails to display image in compiled php

If you have --with-gd try adding --with-jpeg-dir to your configure string.

If you've found a solution to a problem regarding image conversion, please add it to this page.

You are sure, that you have ImageMagick, but it's not working...?

If your **ImageMagick** was successfully detected by the setup guide, but you still don't see images in your site, there might be a path problem.

Edit settings/image.ini and state the complete path **and** executable in Convert**Executable**. It seems as ez 3.0.1 and 3.0.2 do not use the ConvertPath. (*Important: if there is an image.ini.append/.php in /settings/override you have to make changes there instead of the main image.ini!*)

My file looks like this:

```
[ConverterSettings]
UseConvert=true
UseGD=true

[ShellSettings]
ConvertPath=/opt/ImageMagick/bin
ConvertExecutable=/opt/ImageMagick/bin/convert
```

Mixing up node id and object id

Are you sure the ID is an Object ID and not a Node ID? The object ID is the one used in **content/edit** while the node ID is the one used for **content/view**. If you try to fetch object related info by referencing a node ID you will most likely not get what you want.

Example

```
{let test=fetch('content',object,hash(object_id,217))}
{$test|attribute(show,1)}
{/let}
```

217 must be the object number and not the node number. Check which number is used in the admin site if you edit the object.

Not enough PHP memory

Having not enough memory can cause all kinds of problems, which don't necessarily have to lead to the "out of memory" error message. If you get the error message or any kind of weird non-tracable behaviour (page not found, blank screens, etc.) after the initial install, try raising the memory as explained below.

On some configurations, eZ publish 3 needs more than 8Mb internal memory, especially when loaded for the first time.

Please raise the memorylimit for php, i.e. set

```
php_admin_value memory_limit 20M
```

in the (virtual) server config or set

```
memory_limit 20M
```

directly in the PHP configuration file **php.ini**.

You can also try to set it in the PHP file start using the function

```
ini_set ( "memory_limit", "20M" )
```

which sets the value of the given configuration option. Returns the old value on success, FALSE on failure. The configuration option will keep this new value during the script's execution, and will be restored at the script's ending.

If your Webpace is able to handle ".htaccess" files then you could also set it by adding the following line

```
php_value memory_limit 20M
```

which avoid the need to include the previous instruction at the start of ALL PHP scripts.

Note #1. Not all the available options can be changed using ini_set(). This can be changed/set (as of PHP 4.2.0) at level PHP_INI_ALL (Entry can be set anywhere).
Get more detailed info about **ini_set()** at <http://www.php.net/manual/en/function.ini-set.php>

Note #2. This can also be caused by a PHP acellerator not clearing memory.

Open_basedir restriction

I get an 'open_basedir restriction in effect. File is in wrong directory...'. What's wrong?

There are several things that could cause this.

a) This could be the PHP 4.2.3 BUG described below. To solve this, rewrite the Ez3 write routines or convince the provider to upgrade to php 4.3.

More info:

<http://bugs.php.net/bug.php?id=19292>

http://www.faqs.com/knowledge_base/view.phtml/aid/18472

b) It could also mean that you have PHP safe mode on. Try switching safe mode off. Safe mode is a general PHP setting. If you are on a shared environment, you will have to talk to your ISP on this.

Even though the safe mode does not affect the open_basedir directive, safe mode can cause this type of errors.

c) open_basedir wrongly configured.

Change the configuration of open_basedir option in php.ini (see [this thread](#)). Find in the /etc/php.ini file the open_basedir option. If the open_basedir looks like this

```
/path/to/www/root/
```

then this is wrong. It must be the following:

Windows

```
;path:to;www;root;
```

ALL other Operating Systems

```
:path:to:www:root:
```

Sounds strange but it may work (tested with php-4.2.3). You might want to do some backup reading in the [php manual on configuration](#).

Change openbase_dir only for one specific vhost of Apache

You can either choose between a specific directory or no directory (value "none")

```
<VirtualHost 217.160.187.131:80>  
ServerName mydomain.de
```

```
DocumentRoot /home/www/web3/html
User web3
Group webuser
php_admin_value upload_tmp_dir /home/www/web3/phptmp/
php_admin_value open_basedir none
RewriteEngine On
RewriteRule !.(gif|css|jpg|png|jar)$ /index.php
</VirtualHost>
```

Permissions of cache files

Only the superuser can change ownership of a file, when a PHP script creates a file the user which runs Apache is the owner.

However it's possible to set other filepermissions by editing settings/site.ini.php and changing the permission settings in the groups **[FileSettings]** to **0777** for dirs and **0666** for files which allows everyone to read/write them. This allows you to remove the files after they have been created by PHP. *Note:* This is considered a security risk since everyone can read the files created, for instance passwords could be present.

You could also ask the admin of the hoster to add you to the apache group or something similar.

Scripts to help you out

There are two scripts to delete the Cache files anyway: [ClearCache from Hans Tolboom](#) and [one by Ekkehard Doerre](#).

Problems with the translation function

Text translation is automatically disabled when the Locale is set to **eng-GB** in /settings/site.ini(.php).

Process cache enabled reduces performance

Update: as of 3.1 process caching is completed and is the preferred caching for eZ publish.

In version 3.0x process caching is not done yet. So you should leave it off in the configuration.

```
[Templatesettings]
...
ProcessCaching=disabled
```

Once it is done (or at least improved) the processing times of templates will decrease significantly and it can be used.

User site instead of admin site

Try accessing your site this way: <http://yoursite.com/index.php/demo/> (nVH) or <http://yoursite.com/demo/> for the demo site and <http://yoursite.com/index.php/admin/> (nVH) or <http://yoursite.com/admin/> for the admin site.

Alternative: There is an empty entry called DefaultAccess in section SiteSettings in the site.ini.php. Change it to DefaultAccess=demo and you will get the demo site as default view. To access the admin, just append the *admin* as explained above.

For a more indepth explanation of Siteaccess, please take a look at [Configure the site access methods](#).

Warning: Failed to write session data (user).

This can occur if you are using mySQL and you have not set up your **user** and **password** correctly, or you have changed them since you installed eZ Publish.

Typically, the error message you see will look something like this:

```
Warning open(/tmp\sess_bfbb292b1c6af15950270acc789d97ec, O_RDWR)
failed:
No such file or directory (2) in Unknown on line 0

Warning: Failed to write session data (files).
Please verify that the current setting of session.save_path is correct
(/tmp)
in Unknown on line 0
```

To solve the problem, examine your `\settings\override\site.ini.append.php` file to ensure that the values in the **[DatabaseSettings]** section are correct.

What not to do with eZ

This document contains general no-no if you are using eZ. Please add to the list if you do something stupid and want to prevent other people from doing the same.

Users

Setup

Do not remove the main language from Setup -> Translations otherwise you will end up losing all your content as I have.

I moved the Media node (which normally lives under the media tab in the admin) into the content section of the site to see if I could, but I can't get it back to the original location. It still works, it is just not accessible from that tab and it appears on the site

Do not remove the default sections Users and Media otherwise the contextual menus in admin disappear with the links to roles, etc and other associated features aren't shown

If your Locale is not english (eng-GB), do NOT try to install a new version of ezPublish over your existing one. You will probably have some problem to run the setup wizard. Reset your Locale to eng-GB, before you start the setup wizard

Do not open the root node for editing in two separate windows and then press the discard changes button in both windows. I did, and as a result I lost my root node.