



Share your information

eZ Publish Upgrading

©1999 – 2012 eZ Systems AS

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license can be downloaded from <http://www.gnu.org/copyleft/fdl.html>.

Corrections and/or suggestions might be sent to info@ez.no.

This PDF file is generated automatically from the online documentation available at <http://doc.ez.no>.

This version was generated on July 31, 2014.

Contents

1	Upgrading	6
1.1	Releases	9
1.2	How to proceed	11
1.3	Backup and consistency checks	14
1.4	Direct upgrading	16
1.4.1	Direct upgrading to 4.5 from 4.1, 4.2 and 4.3	17
1.4.2	Direct upgrading to 4.6 from 4.1, 4.2, 4.3 and 4.4	42
1.4.3	Direct upgrading to 4.7 from 4.1, 4.2, 4.3, 4.4 and 4.5	78
1.4.4	Direct upgrading to 5.1 from 4.7	126
1.5	Upgrading to 5.1	134
1.5.1	Upgrading from 5.0 to 5.1	135
1.6	Upgrading to 5.0	141
1.6.1	Upgrading from 4.7 to 5.0	142
1.7	Upgrading to 4.7	148
1.7.1	Upgrading from 4.6 to 4.7	149
1.8	Upgrading to 4.6	155
1.8.1	Upgrading from 4.5 to 4.6	156
1.9	Upgrading to 4.5	162
1.9.1	Upgrading from 4.4 to 4.5	163
1.10	Upgrading to 4.4	168
1.10.1	Upgrading from 4.3.x to 4.4.x	169
1.11	Upgrading to 4.3	173
1.11.1	Upgrading from 4.2.x to 4.3.x	174
1.12	Upgrading to 4.2	179

1.12.1	Upgrading from 4.1.x to 4.2.y	180
1.13	Upgrading to 4.1	185
1.13.1	4.1.x to 4.1.y	186
1.13.2	4.0.x to 4.1.y	190
1.14	Upgrading to 4.0	196
1.14.1	Important notes	197
1.14.2	from 4.0.x to 4.0.y	200
1.14.3	from 3.10.x to 4.0.y	212
1.14.4	from 3.10.0 to 4.0.0 (deprecated)	218
1.14.5	from 4.0.0 to 4.0.1rc2 (deprecated)	225
1.15	Upgrading to 3.10	234
1.15.1	from 3.10.x to 3.10.y	235
1.15.2	from 3.9.x to 3.10.y	242
1.15.3	from 3.9.x to 3.10.0 (deprecated)	254
1.16	Upgrading to 3.9	264
1.16.1	from 3.9.x to 3.9.y	265
1.16.2	from 3.8.x to 3.9.0	272
1.17	Upgrading to 3.8	279
1.17.1	from 3.8.x to 3.8.y	280
1.18	Upgrading to 3.7	286
1.18.1	from 3.7.x to 3.7.y	287
1.18.2	from 3.5.2 to 3.7.0	290
1.19	Upgrading to 3.6	295
1.19.1	from 3.6.x to 3.6.y	296
1.19.2	from 3.5.2 to 3.6.0	299
1.20	Upgrading to 3.5	306
1.20.1	from 3.5.x to 3.5.y	307
1.20.2	from 3.4.4 to 3.5.0	311
1.21	Upgrading to 3.4	316
1.21.1	from 3.4.x to 3.4.y	317
1.21.2	from 3.3-5 to 3.4.0	320
1.22	Upgrading to 3.3	324

1.22.1	from 3.3-x to 3.3-y	325
1.22.2	from 3.2-4 to 3.3-1	328
1.23	Upgrading to 3.2	330
1.23.1	from 3.2-x to 3.2-y	331
1.23.2	from 3.1-1 to 3.2-1	334
1.24	Upgrading to 3.1	337
1.24.1	from 3.1-x to 3.1-y	338
1.24.2	from 3.0-2 to 3.1.0 rel1	341
1.25	Upgrading to 3.0	344
1.25.1	from 3.0-1 to 3.0-2	345
1.25.2	from 2.9-7 to 3.0-1	347
1.26	Upgrading from 3.a.b to 3.x.y	350
1.27	The system upgrade scripts	357
1.28	Notes about database changes	373

List of Figures

1.1	All eZ Publish releases and their upgrade paths.	10
-----	--	----

Chapter 1

Upgrading

This chapter contains information related to the upgrade of an eZ Publish solution. If eZ Publish is used correctly, then upgrades should be very easy. This means that if you have not tampered with the kernel files, changed the database, the default settings and/or templates, everything should be okay. If you have, then you should know what you are doing.

The following list contains links to the different upgrade procedures. If you haven't done so already, make sure that you have a backup and that your data is in a consistent state (refer to the text below):

- [How to proceed \(page 11\)](#)
- [Upgrading from 5.1 to 5.2](#)
- [Upgrading from 5.0 to 5.1 \(page 135\)](#)
 - [Direct upgrading from eZ Publish 4.7 to 5.1 \(page 126\)](#)
- [Upgrading from 4.7 to 5.0 \(page 142\)](#)
- [Upgrading from 4.6 to 4.7 \(page 149\)](#)
 - [Direct upgrading from 4.1, 4.2, 4.3, 4.4 and 4.5 to 4.7 \(page 78\)](#)
- [Upgrading from 4.5 to 4.6 \(page 156\)](#)
 - [Direct upgrading from eZ Publish 4.1, 4.2, 4.3 and 4.4 to 4.6 \(page 42\)](#)
- [Upgrading from 4.4 to 4.5 \(page 163\)](#)
 - [Direct upgrading from eZ Publish 4.1, 4.2 and 4.3 to 4.5 \(page 17\)](#)
- [Upgrading from 4.3 to 4.4 \(page 169\)](#)
- [Upgrading from 4.2 to 4.3 \(page 174\)](#)
- [Upgrading from 4.1.x to 4.2 \(page 180\)](#)

- Upgrading from 4.1.x to 4.1.y (page [190](#))
- Upgrading from 4.0.x to 4.1.y (page [190](#))
- Notes about upgrading to 4.0 (page [197](#))
- Upgrading from 4.0.x to 4.0.y (page [200](#))
- Upgrading from 3.10.x to 4.0.y (page [212](#))
- Upgrading from 3.10.x to 3.10.y (page [235](#))
- Upgrading from 3.9.x to 3.10.y (page [242](#))
- Upgrading from 3.9.x to 3.9.y (page [265](#))
- Upgrading from 3.8.x to 3.9.0 (page [272](#))
- Upgrading from 3.8.x to 3.8.y (page [280](#))
- Upgrading from 3.6.x or 3.7.x to 3.8.0
- Upgrading from 3.7.x to 3.7.y (page [287](#))
- Upgrading from 3.5.2 to 3.7.0 (page [290](#))
- Upgrading from 3.6.x to 3.6.y (page [296](#))
- Upgrading from 3.5.2 to 3.6.0 (page [299](#))
- Upgrading from 3.5.x to 3.5.y (page [307](#))
- Upgrading from 3.4.4 to 3.5.0 (page [311](#))
- Upgrading from 3.4.x to 3.4.y (page [317](#))
- Upgrading from 3.3-5 to 3.4.0 (page [320](#))
- Upgrading from 3.3-x to 3.3-y (page [325](#))
- Upgrading from 3.2-4 to 3.3-1 (page [328](#))
- Upgrading from 3.2-x to 3.2-y (page [331](#))
- Upgrading from 3.1-1 to 3.2-1 (page [334](#))
- Upgrading from 3.1-x to 3.1-y (page [338](#))
- Upgrading from 3.0-2 to 3.1.0 rel1 (page [341](#))
- Upgrading from 3.0-1 to 3.0-2 (page [345](#))
- Upgrading from 2.9.7 to 3.0-1 (page [347](#))
- Upgrading from 3.a.b to 3.x.y (page [350](#))

- The system upgrade scripts (for eZ Publish 3.x) (page [357](#))
- The upgrade path from one branch to another. (page [9](#))
- Notes about database changes (page [373](#))

1.1 Releases

Branches

eZ Publish consists of multiple branches, each branch defines a set of features which is constant throughout the branch. A branch consists of multiple releases.

Releases

Alpha releases are created during the implementation of new features to allow early testing. These often lack features promised in the final release and are considered quite unstable.

Beta releases are created once all the new features have been implemented. A beta version is likely to be unstable. The testers of these releases report any bugs that they found and sometimes minor features they would like to see in the final version.

A release candidate is a final product, ready to release unless fatal bugs emerge. At this stage, all the new features are implemented and tested, all known important bugs are fixed.

Stable releases are created after the branch is considered stable which means that all major bugs and hurdles have been removed. Generally, the database schema is not changed within stable release branches. However, exceptions are made for changes that are needed for important bugs and enhancements.

Overview

The following diagram shows the various branches and releases that have been made for eZ Publish and indicates the upgrade path from one branch to another.

(see figure 1.1)

Upgrades

The arrows show the upgrade paths that are supported by eZ Publish. If you have a stable release which does not have a direct upgrade path, you can use the upgrade path from a previous stable release in the same branch. Note that you might have to skip some parts of the database upgrade scripts if the database schema was changed from previous stable releases within the same branch. For more information about which parts to skip, refer to the upgrade instructions of the version to which you are upgrading.

Share information

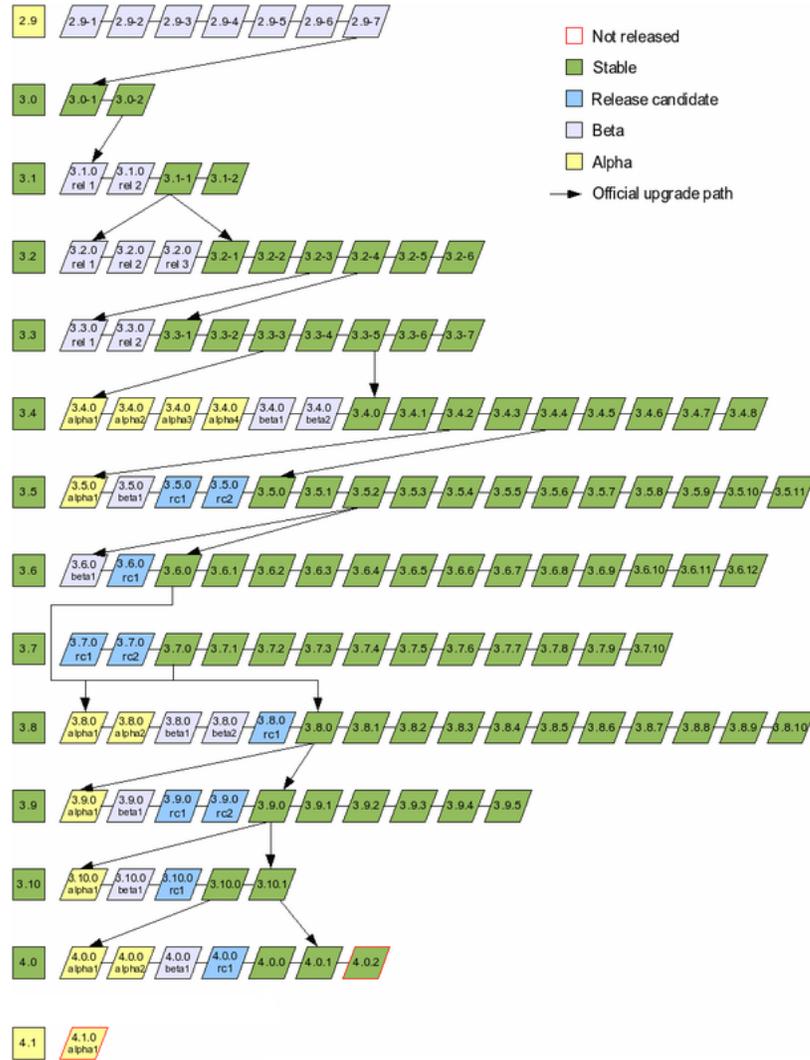


Figure 1.1: All eZ Publish releases and their upgrade paths.

1.2 How to proceed

This section describes how to upgrade your existing eZ Publish installation to a newer version.

If you are running one of the [supported versions](#), it is recommended to upgrade your installation when the next stable release within that branch is out. The instructions for upgrading within the same stable release branch can be found in the upgrade manuals of that branch.

If you are running a version which is no longer supported, it is highly recommended to switch to a supported one. For example, if you are running eZ Publish 4.1, consider upgrading to version 4.5 or later. In order to benefit from the latest security and bug fixes, choose the latest stable release in the desired branch. You can either upgrade directly or split the upgrade process into several stages (depending on the number of stable release branches between your current version and the desired one).

Direct upgrade

If you want to upgrade directly from version X to version Y, the upgrade procedure consists of the following steps:

1. Upgrading the distribution files to version Y.
2. Upgrading the database schema to version Y following the official upgrade path.
3. Running the system upgrade scripts for versions between X and Y.
4. Updating the system configuration for compatibility with all the changes made from version X to Y.
5. Clearing the caches.

This procedure is only applicable when upgrading to the next stable release branch. Skipping a branch is not recommended, since some of the system upgrade scripts might not be present (or not work correctly) in later versions. The only exception is that you can upgrade directly from eZ Publish 3.6 to version 3.8.

The instructions for direct upgrading from version 3.a.b to 3.x.y can be found here (page [350](#)). In order to upgrade from version 3.10 to 4.0, refer to the eZ Publish 4.0 upgrade manuals (page [196](#)).

Two upgrades

If a lot of changes were made between versions X and Y, you can do the following:

1. Upgrade from version X to some intermediate version Z.
2. Make sure that your site is working correctly.
3. Upgrade from version Z to version Y.

Example 1

If you are running eZ Publish 3.8.x, you cannot upgrade directly to 3.10.y, because the system upgrade scripts needed for eZ Publish 3.9 might not work correctly in 3.10. However, you can upgrade to some intermediate version 3.9.z first and then upgrade to version 3.10.y.

Example 2

It is not possible to upgrade directly from eZ Publish 3.9 to version 4.0, and the 4.0 distribution does not contain upgrade scripts for this. The solution is to upgrade to 3.10 first, and then upgrade to 4.0.

Multiple upgrades

It is also possible to split the upgrade process into three or more stages. Since each branch introduces a set of new features, you have to do one upgrade for each stable release branch between versions X and Y. For example, if you are running eZ Publish 3.7 and want to run 3.10, you have to upgrade to 3.8 first, then to 3.9 and then finally to 3.10.

While the last direct upgrade in the chain should be done to the latest stable release in the desired branch, there are two different rules about which versions to choose for preceding upgrades. You can either follow the official upgrade path for database schema changes or select the latest stable release in each branch. The following text describes the advantages and disadvantages of both approaches.

The "database upgrade path" approach

This approach makes it possible to minimize the number of database upgrade scripts involved and avoid situations where you need to skip some parts in the ".sql" files when upgrading the database. In the example above, the official upgrade path for database changes looks like this:

```
3.7 -> 3.8.0 -> 3.9.0 -> 3.10
```

This means that you should upgrade to 3.8.0 first, then to 3.9.0 and then finally to 3.10.

The disadvantage of this approach is that you upgrade to early versions in each branch, which may contain bugs that were fixed in later releases. (Important bugs are mentioned in the upgrade instructions for certain versions of eZ Publish. In addition, changelogs provide information about all issues that were fixed in the latest releases.)

If an early version contains a bug that may corrupt your data, you can either fix the problem yourself (if possible) or upgrade to a later release where this issue was fixed. For example, when upgrading from 3.8 to 3.9 you need to run the "correctxmltext.php" upgrade script. If you run this script when upgrading to 3.9.0, it will crash due to a bug. To make it work, you need to edit the script file manually as described [here](#). Another option is to skip version 3.9.0 and

upgrade directly to version 3.9.1 or a later 3.9.x release where the updated version of the script is available.

The "latest stable release" approach

Upgrading directly to the latest stable release in every branch between X and Y allows you to avoid dealing with possible bugs that might be present in earlier versions. With this approach, the path for upgrading from eZ Publish 3.7 to version 3.10 would look like this:

```
3.7 -> 3.8.10 -> 3.9.4 -> 3.10
```

This means that you should upgrade to 3.8.10 first, then to 3.9.4 (the latest stable release in the 3.9.x branch at the time of writing) and then finally to 3.10.

Generally, this approach requires running more database upgrade scripts than the previous one (since there is usually one database upgrade script for each stable release). However, you can reduce the number of database upgrade scripts to run by skipping all version-focused scripts for intermediary versions.

The disadvantage of the "latest stable release" approach is that you might have to skip some parts of the database upgrade scripts if the database schema was changed from previous stable releases within the same branch. This is because the upgrade scripts between branches contain all of the necessary schema changes from the first stable release in the older branch to the first stable release in the newer branch. (Refer to "Notes about database changes" for more information about the database schema changes within stable release branches.)

For example, in eZ Publish 3.8.5, some database schema changes were introduced as a bug fix for [missing indexes in database tables](#). The fix was added to both eZ Publish 3.8.5 and 3.9.0 beta1 and included in the following database upgrade scripts:

- .../3.8/dbupdate-3.8.4-to-3.8.5.sql
- .../3.9/unstable/dbupdate-3.9.0alpha1-to-3.9.0beta1.sql
- .../3.9/dbupdate-3.8.0-to-3.9.0.sql

If you are upgrading to eZ Publish 3.9 from version 3.8.5 or later, running the "dbupdate-3.8.0-to-3.9.0.sql" upgrade script will produce an error, because your 3.8.5 database already contains some of the database schema changes included in this ".sql" file. The solution is to skip the part marked with "from 3.8.5" comments in this file.

1.3 Backup and consistency checks

Backup

Before you do anything else, you should make sure that the current solution is properly backed up. In other words, you should create a copy of the entire eZ Publish directory and the database. The following example shows how this can be done on a Linux/UNIX based system where eZ Publish is using a MySQL database called "example". Note that the system should be closed for access during backups and upgrades.

1. Navigate into the eZ Publish directory:

```
$ cd /path/to/ez_publish
```

2. Clear all caches:

```
$ bin/shell/clearcache.sh --clear-all
```

 If you have [PHP CLI](#) installed, you can also do the following (this script should also work on Windows):

```
$ ./bin/php/ezcache.php --clear-all --purge
```

 From 3.10, the "clearcache.sh" script has been disabled. This means that the file is still present, but the code inside is removed. It outputs a message telling the user to use "bin/php/ezcache.php". In other words, you have to use "ezcache.php" instead. After running the script, make sure that all cache files have been cleared by inspecting the contents of the various cache sub-directories within the "var" directory (typically the "var/cache/" and "var/<name_of_siteaccess>/cache/" directories). If there are any cache files left, remove them manually.

3. Create a dump of the database:

```
$ mysqldump -u root --add-drop-table example > db_backup.sql
```

4. Go to the parent directory:

```
$ cd ..
```

5. Create a tape archive of the files (including the DB dump) using the "tar" command:

```
$ tar cfz backup_of_ez_publish.tar.gz ez_publish
```

 At this point, the file "backup_of_ez_publish.tar.gz" should contain a backup of everything (both files and DB).

Consistency checks

The administration interface makes it possible to quickly check whether the current installation is in a consistent state or not. This can be done using the "Upgrade check" section of the "Setup" part. There are two checks that can be run:

- File consistency check
- Database consistency check

File consistency check

The file consistency tool checks if you have altered any of the files that came with the current installation. If this is the case, then the altered files should be backed up before the system is upgraded because they will most likely be overwritten by new versions. Make sure that you backup and then merge in your custom changes into the new versions of the files.

Database consistency check

The database consistency feature checks if the current database is consistent with the database schema that came with the eZ Publish distribution that is currently running. If there are any inconsistencies, the system will suggest the necessary SQL statements that should be ran in order to bring the database into a consistent state. Make sure that the database is backed up and run the suggested SQL statements before upgrading.

Note: If your installation does not have the eZ Network installed, you will get an error message about database inconsistency. This is to be expected and the error message will disappear as soon as the eZ Network is installed. This applies to eZ Publish 4.5 and above.

Note2: If using cluster with the same database as eZ Publish then you will get error message about database inconsistency (specifically: ezdbfile* or ezdfsfile tables). This is a known issue with database inconsistency check combined with clustering, and can safely be ignored. To avoid this you can install cluster tables in a separate database, and make sure this is reflected in config.php and file.ini settings, see "Features > Clustering" for generic information about how to install cluster.

The actual upgrade process

An actual eZ Publish upgrade process is usually carried out in a few steps. First, the current eZ Publish files must be replaced with the files from the new distribution. Secondly, a couple of scripts must be run in order to bring the database (and possibly other things) into a state that is compatible with the upgraded files. In addition, it might be needed to change the system configuration (review some INI settings, add templates, etc.). Refer to the "How to proceed (page 11)" section for general information about these steps.

1.4 Direct upgrading

Follow the upgrading procedures below for easier, step by step upgrading from earlier versions to the latest versions of eZ Publish:

- Direct upgrading to 5.1 from 4.7 (page [126](#))
- Direct upgrading to 4.7 from 4.1, 4.2, 4.3, 4.4, 4.5, 4.6 (page [78](#))
- Direct upgrading to 4.6 from 4.1, 4.2, 4.3, 4.4 (page [42](#))
- Direct upgrading to 4.5 from 4.1, 4.2, 4.3 (page [17](#))

1.4.1 Direct upgrading to 4.5 from 4.1, 4.2 and 4.3

Please note that there are various documentation options for the direct upgrading to eZ Publish 4.5. There are 3 separate upgrade kits containing the information for upgrading from versions 4.1, 4.2 and 4.3 to 4.5 of eZ Publish.

- Direct upgrading from 4.1 to 4.5 (page [18](#))
- Direct upgrading from 4.2 to 4.5 (page [27](#))
- Direct upgrading from 4.3 to 4.5 (page [35](#))

Direct upgrading from 4.1 to 4.5

This section describes how to upgrade your existing eZ Publish 4.1 installation directly to version 4.5. Make sure that you have a working backup of the site before you do the actual upgrade, and make sure the installation you are performing the upgrade on is offline.

Important upgrade notes:

The procedure for upgrading directly from version 4.1.x, 4.2 or 4.3 to 4.5 consists of the following steps:

- Upgrade the distribution files to 4.5
- Upgrade custom extensions
 - 4.1.0 to 4.2
 - 4.2 to 4.3
 - 4.3 to 4.4
 - 4.4 to 4.5
- Upgrade the database to 4.5
 - 4.1.0 to 4.2
 - 4.2 to 4.3
 - 4.3 to 4.4
 - 4.4 to 4.5
- Regenerate the autoload array for extensions
- Run the system upgrade scripts for 4.5
 - 4.1.0 to 4.2
 - 4.2 to 4.3
 - 4.3 to 4.4
 - 4.4 to 4.5
- Clear the caches

Check for requirements

The eZ Components and PHP requirements

The minimum version required of eZ Components with eZ Publish 4.5 "ezcomponents-ezp45", containing a fix to the package. This version is bundled in eZ Publish 4.5 Enterprise. eZ Publish 4.5 is compatible with PHP version 5.2.1 and above, but certified on RHEL 6 and Debian 6, PHP 5.3.x distros. See the requirements page for more info.

Step 1: Upgrade the distribution files

The easiest way to upgrade the distribution files is to unpack eZ Publish 4.5 to a separate directory and then copy the directories that contain site-specific files from the existing installation. Make sure that you copy the following directories:

- design/example
- design/example_admin
- var
- settings/siteaccess
- settings/override
- (index_cluster.php)

Cluster note: If you are using cluster mode its necessary to copy index_cluster.php from the old version.

Replace "example" and "example_admin" with the actual names of your site accesses.

Important note: Because the new directory has replaced the original directory, the directory permissions need to be fixed. Use the following commands to do this.

(You have the choice between Shell commands or Alternative shell commands):

- **Shell commands**

These shell commands will give proper permission to the web server:

```
cd </path/to/your/eZ/Publish/directory>
chmod -R a+rx design extension settings var
```

- **Alternative shell commands**

These commands will setup the permission more correctly, but require knowledge about the running web server (change "nouser" to http server user).

```
cd </path/to/your/eZ/Publish/directory>
chmod -R og+rx design extension settings var
chown -R nouser:nouser design extension settings var
```

Step 2: Custom extensions

If you are using custom extensions, the sub-directories inside the "extension" directory will also have to be copied. However, make sure that you do not overwrite any extensions that are included in eZ Publish, which currently are eZ Flow (2.x), eZ Online Editor (5.x), eZ OpenOffice Document format (2.x), eZ JSCore (1.x), eZ Image Editor (1.x), eZ Comments (1.x), eZ Multiupload (1.x), eZ MB Password Expiry (1.x), eZ Network (1.x), eZ REST API Provider (1.x), eZ Script Monitor (1.x), eZ SI, eZ Find (2.x). Note that upgrading the distribution files will overwrite the autoload arrays for extensions. You will need to re-generate the autoload arrays for active extensions later.

See the dedicated upgrade instructions for eZ Flow and Website Interface below.

The updated versions of eZ Flow and Website Interface will also install the following extensions:

- eZ Website Toolbar
- eZ Star Rating
- eZ Google Maps Location

Note: For eZ Online Editor 5.x and eZ JS Core you will need to replace the following rewrite rules when using Virtual Hosts:

```
RewriteRule ^/var/cache/texttoimage/./* - [L]
RewriteRule ^/var/[~/]+/cache/(texttoimage|public)/.* - [L]
```

with:

```
RewriteRule ^/var/([~/]+)?cache/(texttoimage|public)/.* - [L]
```

For more detailed instructions, see the dedicated [eZ OE](#) and eZ JS Core doc pages.

Step 3: updateimagesystem.php

This script is necessary to run if you have data of the "ezimage" datatype dating from before eZ Publish 3.3. In version 3.3 a new image system took over based on "ezimagealiashandler".

If you are upgrading to the 4.2 series of eZ Publish for the first time, and the installation has been running since prior to eZ Publish 3.3 then you need to run the "updateimagesystem.php" script before running any of the "dbupdate" scripts.

If the installation is a new installation or based on any eZ Publish version since 3.3 then you can skip running the "updateimagesystem.php" script.

If not, then execute it using:

```
php update/common/scripts/4.1/updateimagesystem.php -s $SITE_ACCESS
```

Step 4: Upgrade the database

The update script for the database is located in the following locations, skip the ones that don't apply to the version your upgrading from:

```
<eZ Publish root>/update/database/<mysql|postgresql>/4.2/
dbupdate-4.1.0-to-4.2.0.sql
<eZ Publish root>/update/database/<mysql|postgresql>/4.3/
dbupdate-4.2.0-to-4.3.0.sql
<eZ Publish root>/update/database/<mysql|postgresql>/4.4/
dbupdate-4.3.0-to-4.4.0.sql
<eZ Publish root>/update/database/<mysql|postgresql>/4.5/
dbupdate-4.4.0-to-4.5.0.sql
```

You can run these with the appropriate sql command line tool or application.

Note on 4.1.x to 4.2 upgrade: You need to selectively choose the correct sql from first file above if your upgrading from 4.1.1, 4.1.2, 4.1.3 or 4.1.4.

If your upgrading from 4.1.1: Ignore everything above -- END: from 4.1.1.

If your upgrading from 4.1.2 or 4.1.3: Ignore everything above -- END: from 4.1.2.

If your upgrading from 4.1.4: Ignore everything above -- END: from 4.1.4.

Clustering:

If your upgrading from 4.1 or 4.2 using db or dfs cluster, make sure you apply the following sql updates (but for dfs make sure you change table name from "ezdbfile" to "ezdfsfile"):

```
<eZ Publish root>/update/database/mysql/4.3/dbupdate-cluster-4.2.0-to-4.3.0.sql
```

Schema changes

Note: These schema changes are only necessary to do if you have eZ Flow installed and if you are making the following upgrades:

4.1.1 -> 4.5

4.1.2 -> 4.5

4.1.3 -> 4.5

4.1.4 -> 4.5

4.2 -> 4.5

you must run this sql on the mysql client

```
ALTER TABLE ezm_pool ADD INDEX ezm_pool__block_id__ts_hidden ( block_id,
ts_hidden );
```

MySQL

The "ezmysql" database driver has been deprecated in 4.5 and it is recommended to use the "ezmysqli" driver instead. This requires that the "mysqli" extension in PHP is enabled and can be archived by changing the driver in your override or siteaccesses "site.ini.append.php" settings to:

```
[DatabaseSettings]
DatabaseImplementation=ezmysqli
```

Cluster

The "Mysql DB/DFS Cluster" database back-end has not yet been deprecated, but it is recommended to use the same driver as the main database driver, so change your "file.ini.append.php" settings in override or siteaccesses to use "eZDFSFileHandlerMySQLiBackend" for DFS Cluster or "eZDBFileHandlerMysqliBackend" for DB Cluster.

Note: The "ndex_cluster.php" file should be updated to

```
define( 'STORAGE_BACKEND', 'dfsmysqli' )
```

If you use DB cluster, remove "dfs" from the string.

Step 5: Regenerate the autoload array for extensions

The autoload system also has some changes, for example the autoload array for extensions is now placed in var/autoload of your eZ Publish installation (along the class changes in extensions itself).

To regenerate the autoload array, execute the following script from the root of your eZ Publish directory:

```
php bin/php/ezpgenerateautoloads.php --extension
```

Step 6: Run the system upgrade scripts

4.1 to 4.2

- `correctxmlalign.php`: after upgrading to 4.1 the default alignment for embedded images in the "ezxmltext" datatype is removed, this script will allow you to have the correct xmlalignment.

```
php update/common/scripts/4.1/correctxmlalign.php -s $SITE_ACCESS
```

- `fixorphanimages.php`: this script will delete images from the storage directory that are no longer connected to any content object attribute

```
php update/common/scripts/4.2/fixorphanimages.php -s $SITE_ACCESS
```

4.2 to 4.3

This version includes a script to fix #15478: Node assignment is not removed when removing node from child list:

```
php update/common/scripts/4.3/updatesnodeassignment.php -s $SITE_ACCESS
```

4.3 to 4.4

Update script for this version contained a bug and is repeated in 4.4 to 4.5 bellow, thus is not needed in this instruction.

4.4 to 4.5

For eZ Publish 4.4 which was upgraded from 4.3, run the following command from the root of your eZ Publish directory:

```
php update/common/scripts/4.5/updatesectionidentifier.php -s $SITE_ACCESS
```

Note: Skip this step if this your 4.4 installation is not an upgrade from 4.3. If you are uncertain, please run the script.

Step 7: Enable ezjscore

Both the "admin2" design, "ezoe", "ezwebin", "ezflow" and several other extensions requires the ezjscore extension to be enabled, this is archived by editing your "settings/override/site.ini.append.php" settings file to make sure it looks like this:

```
<pre>
[ExtensionSettings]
ActiveExtensions []
ActiveExtensions []=ezjscore
</pre>
```

Step 8: Enable admin2 design for back-end

To enable admin2 open "site.ini" for your admin site access. For an eZ Flow site this would for instance be:

```
settings/siteaccess/ezflow_site_admin/site.ini.append.php
```

In this file include "AdditionalSiteDesignList[]=admin2" to your [DesignSettings]. It is important that you add the *admin2* design before the *admin* design, in order to let extensions that have admin templates to continue to work. On an eZ Flow site, it would mean changing the settings from:

```
[DesignSettings]
SiteDesign=ezflow_site_admin
AdditionalSiteDesignList[]=admin
AdditionalSiteDesignList[]=ezflow
```

to

```
[DesignSettings]
SiteDesign=ezflow_site_admin
AdditionalSiteDesignList[]=admin2
AdditionalSiteDesignList[]=admin
AdditionalSiteDesignList[]=ezflow
```

When using a plain or eZ Webin install you should make the same changes. In this case you will of course have a different SiteDesign setting and no AdditionalSiteDesignList[]=ezflow line.

To enable users to change their Administration Interface preferences (Location and Re-Edit editing preferences), add the following line to the [Toolbar_admin_right] block in settings/siteaccess/<siteaccess_name>/toolbar.ini.append.php:

```
Tool[]=admin_preferences
```

So the block will then look like like this:

```
[Toolbar_admin_right]
Tool[]
Tool[]=admin_current_user
Tool[]=admin_bookmarks
Tool[]=admin_preferences
```

Step 9: Clear the caches

Whenever an eZ Publish solution is upgraded, all caches must be cleared in a proper way. **Note:** You should not be a root user when you start the caching process (refer to <http://issues.ez.no/15823>). The caching should be done from within a system shell:

1. Navigate into the eZ Publish 4.4 directory.
2. Run the script using the following shell command:

```
php bin/php/ezcache.php --clear-all --purge
```

Purging ensures that the caches are physically removed. When the "--purge" parameter is not specified, the caches will be expired but not removed.

Sometimes the script is unable to clear all cache files because of restrictive file/directory permission settings. Make sure that all cache files have been cleared by inspecting the contents of the various cache sub-directories within the "var" directory (typically the "var/cache/" and "var/<name_of_siteaccess>/cache/" directories). If there are any cache files left, you need to remove them manually.

Step 10: Upgrade extensions

When using eZ Webin and eZ Flow, these extension will also need to be updated.

The first thing to do is to change the extensions permissions. An example on how this is done:

```
$chmod -R a+rwX extension
```

4.1 to 4.2:

eZ Webin 1.4 to 1.5

or

eZ Flow 1.1 to 2.0

4.2 to 4.3:

eZ Webin 1.5 to 1.6

or

eZ Flow 2.0 to 2.1

4.3 to 4.4:

eZ Webin 1.6 to 1.7

or

eZ Flow 2.1 to 2.2

4.4 to 4.5:

eZ Webin 1.7 to 1.8

or
eZ Flow 2.2 to 2.3

Share your information

Direct upgrading from 4.2 to 4.5

This section describes how to upgrade your existing eZ Publish 4.2 installation directly to version 4.5. Make sure that you have a working backup of the site before you do the actual upgrade, and make sure the installation you are performing the upgrade on is offline.

Important upgrade notes:

The procedure for upgrading directly from version 4.2 to 4.5 consists of the following steps:

- Upgrade the distribution files to 4.5
- Upgrade custom extensions
 - 4.2 to 4.3
 - 4.3 to 4.4
 - 4.4 to 4.5
- Upgrade the database to 4.5
 - 4.2 to 4.3
 - 4.3 to 4.4
 - 4.4 to 4.5
- Regenerate the autoload array for extensions
- Run the system upgrade scripts for 4.5
 - 4.2 to 4.3
 - 4.3 to 4.4
 - 4.4 to 4.5
- Clear the caches

Check for requirements

The eZ Components and PHP requirements

The minimum version required of eZ Components with eZ Publish 4.5 is "ezcomponents-ezp45", containing a fix to the package. This version is bundled in eZ Publish 4.5 Enterprise. eZ Publish 4.5 is compatible with PHP version 5.2.1 and above, but certified on RHEL 6 and Debian 6, PHP 5.3.x distros. See <http://ez.no/ezpublish/requirements> for more info.

Step 1: Upgrade the distribution files

The easiest way to upgrade the distribution files is to unpack eZ Publish 4.5 to a separate directory and then copy the directories that contain site-specific files from the existing installation. Make sure that you copy the following directories:

- design/example
- design/example_admin
- var
- settings/siteaccess
- settings/override
- (index_cluster.php)

Cluster note: If you are using cluster mode its necessary to copy index_cluster.php from the old version.

Replace "example" and "example_admin" with the actual names of your site accesses.

Important note: Because the new directory has replaced the original directory, the directory permissions need to be fixed. Use the following commands to do this.

(You have the choice between Shell commands or Alternative shell commands):

- **Shell commands**

These shell commands will give proper permission to the web server:

```
cd </path/to/your/eZ/Publish/directory>
chmod -R a+rx design extension settings var
```

- **Alternative shell commands**

These commands will setup the permission more correctly, but require knowledge about the running web server (change 'nouser' to http server user).

```
cd </path/to/your/eZ/Publish/directory>
chmod -R og+rx design extension settings var
chown -R nouser:nouser design extension settings var
```

Step 2: Custom extensions

If you are using custom extensions, the sub-directories inside the "extension" directory will also have to be copied. However, make sure that you do not overwrite any extensions that are included in eZ Publish, which currently are eZ Flow (2.x), eZ Online Editor (5.x), eZ OpenOffice Document format (2.x), eZ JSCore (1.x), eZ Image Editor (1.x), eZ Comments (1.x), eZ Multiupload (1.x), eZ MB Password Expiry (1.x), eZ Network (1.x), eZ REST API Provider (1.x), eZ Script Monitor (1.x), eZ SI, eZ Find (2.x). Note that upgrading the distribution files will overwrite the autoload arrays for extensions. You will need to re-generate the autoload arrays for active extensions later.

See the dedicated upgrade instructions for eZ Flow and Website Interface below.

The updated versions of eZ Flow and Website Interface will also install the following extensions:

- eZ Website Toolbar
- eZ Star Rating
- eZ Google Maps Location

Note: For eZ Online Editor 5.x and eZ JS Core you will need to replace the following rewrite rules when using Virtual Hosts:

```
RewriteRule ^/var/cache/texttoimage/./* - [L]
RewriteRule ^/var/[^/]+/cache/(texttoimage|public)/.* - [L]
```

with:

```
RewriteRule ^/var/([~/]+)?cache/(texttoimage|public)/.* - [L]
```

For more detailed instructions, see the dedicated [eZ OE](#) and eZ JS Core doc pages.

Step 3: updateimagesystem.php

If the installation is a new installation or based on any eZ Publish version since 3.3 then you can skip running the "updateimagesystem.php" script. If not, then execute it using:

```
php update/common/scripts/4.1/updateimagesystem.php -s $SITE_ACCESS
```

Step 4: Upgrade the database

The update script for the database is located in the following locations, skip the ones that don't apply to the version your upgrading from:

```
<eZ Publish root>/update/database/<mysql|postgresql>/4.3/
dbupdate-4.2.0-to-4.3.0.sql
<eZ Publish root>/update/database/<mysql|postgresql>/4.4/
dbupdate-4.3.0-to-4.4.0.sql
<eZ Publish root>/update/database/<mysql|postgresql>/4.5/
dbupdate-4.4.0-to-4.5.0.sql
```

You can run these with the appropriate sql command line tool or application.

Clustering:

If your upgrading from 4.1 or 4.2 using db or dfs cluster, make sure you apply the following sql updates (but for dfs make sure you change table name from "ezdbfile" to "ezdfsfile"):

```
<eZ Publish root>/update/database/mysql/4.3/dbupdate-cluster-4.2.0-to-4.3.0.sql
```

Schema changes

Note: The schema changes are only necessary if you have eZ Flow installed and if you are making an upgrade from 4.2 -> 4.5. If so you must run this sql on the mysql client

```
ALTER TABLE ezm_pool ADD INDEX ezm_pool__block_id__ts_hidden ( block_id,
ts_hidden );
```

MySQL

The "ezmysql" database driver has been deprecated in 4.5 and it is recommended to use the "ezmysqli" driver instead. This requires that the "mysqli" extension in PHP is enabled and can be archived by changing the driver in your override or siteaccesses "site.ini.append.php" settings to:

```
[DatabaseSettings]
DatabaseImplementation=ezmysqli
```

Cluster

The "Mysql DB/DFS Cluster" database back-end has not yet been deprecated, but it is recommended to use the same driver as the main database driver, so change your "file.ini.append.php" settings in override or siteaccesses to use "eZDFSFileHandlerMySQLiBackend" for DFS Cluster or "eZDBFileHandlerMysqliBackend" for DB Cluster.

Note: The "index_cluster.php" file should be updated to

```
define( 'STORAGE_BACKEND', 'dfsmysqli' )
```

If you use DB cluster, remove "dfs" from the string.

Step 5: Regenerate the autoload array for extensions

The autoload system also has some changes, for example the autoload array for extensions is now placed in var/autoload of your eZ Publish installation (along the class changes in extensions itself).

To regenerate the autoload array, execute the following script from the root of your eZ Publish directory:

```
php bin/php/ezpgenerateautoloads.php --extension
```

Step 6: Run the system upgrade scripts

4.2 to 4.3

This version includes a script to fix #15478: Node assignment is not removed when removing node from child list:

```
php update/common/scripts/4.3/updatenodeassignment.php -s $SITE_ACCESS
```

4.3 to 4.4

Update script for this version contained a bug and is repeated in 4.4 to 4.5 bellow, thus is not needed in this instruction.

4.4 to 4.5

For eZ Publish 4.4 which was upgraded from 4.3, run the following command from the root of your eZ Publish directory:

```
php update/common/scripts/4.5/updatesectionidentifier.php -s $SITE_ACCESS
```

Note: Skip this step if this your 4.4 installation is not an upgrade from 4.3. If you are uncertain, please run the script.

Step 7: Enable ezjscore

Both the "admin2" design, "ezoe", "ezwebin", "ezflow" and several other extensions requires the ezjscore extension to be enabled, this is archived by editing your "settings/override/site.ini.append.php" settings file to make sure it looks like this:

```
<pre>
[ExtensionSettings]
ActiveExtensions []
ActiveExtensions []=ezjscore
</pre>
```

Step 8: Enable admin2 design for back-end

To enable admin2 open "site.ini" for your admin site access. For an eZ Flow site this would for instance be:

```
settings/siteaccess/ezflow_site_admin/site.ini.append.php
```

In this file include "AdditionalSiteDesignList[]=admin2" to your [DesignSettings]. It is important that you add the *admin2* design before the *admin* design, in order to let extensions that have admin templates to continue to work. On an eZ Flow site, it would mean changing the settings from:

```
[DesignSettings]
SiteDesign=ezflow_site_admin
AdditionalSiteDesignList []=admin
AdditionalSiteDesignList []=ezflow
```

to

```
[DesignSettings]
SiteDesign=ezflow_site_admin
AdditionalSiteDesignList []=admin2
AdditionalSiteDesignList []=admin
AdditionalSiteDesignList []=ezflow
```

When using a plain or eZ Webin install you should make the same changes. In this case you will of course have a different SiteDesign setting and no AdditionalSiteDesignList[]=ezflow line.

To enable users to change their Administration Interface preferences (Location and Re-Edit editing preferences), add the following line to the [Toolbar_admin_right] block in settings/siteaccess/<siteaccess_name>/toolbar.ini.append.php:

```
Tool []=admin_preferences
```

So the block will then look like like this:

```
[Toolbar_admin_right]
Tool []
Tool []=admin_current_user
Tool []=admin_bookmarks
Tool []=admin_preferences
```

Step 9: Clear the caches

Whenever an eZ Publish solution is upgraded, all caches must be cleared in a proper way. **Note:** You should not be a root user when you start the caching process (refer to <http://issues.ez.no/15823>). The caching should be done from within a system shell:

1. Navigate into the eZ Publish 4.4 directory.
2. Run the script using the following shell command:

```
php bin/php/ezcache.php --clear-all --purge
```

Purging ensures that the caches are physically removed. When the "--purge" parameter is not specified, the caches will be expired but not removed.

Sometimes the script is unable to clear all cache files because of restrictive file/directory permission settings. Make sure that all cache files have been cleared by inspecting the contents of the various cache sub-directories within the "var" directory (typically the "var/cache/" and "var/<name_of_siteaccess>/cache/" directories). If there are any cache files left, you need to remove them manually.

Step 10: Upgrade extensions

When using eZ Webin and eZ Flow, these extension will also need to be updated.

The first thing to do is to change the extensions permissions. An example on how this is done:

```
$chmod -R a+rx extension
```

4.2 to 4.3:

eZ Webin 1.5 to 1.6

or

eZ Flow 2.0 to 2.1

4.3 to 4.4:

eZ Webin 1.6 to 1.7

or

eZ Flow 2.1 to 2.2

4.4 to 4.5:

eZ Webin 1.7 to 1.8

or

eZ Flow 2.2 to 2.3

Share your information

Direct upgrading from 4.3 to 4.5

This section describes how to upgrade your existing eZ Publish 4.3 installation directly to version 4.5. Make sure that you have a working backup of the site before you do the actual upgrade, and make sure the installation you are performing the upgrade on is offline.

Important upgrade notes:

The procedure for upgrading directly from version 4.3 to 4.5 consists of the following steps:

- Upgrade the distribution files to 4.5
- Upgrade custom extensions
 - 4.3.to 4.4
 - 4.4 to 4.5
- Upgrade the database to 4.5
 - 4.3 to 4.4
 - 4.4 to 4.5
- Regenerate the autoload array for extensions
- Run the system upgrade scripts for 4.5
 - 4.3 to 4.4
 - 4.4 to 4.5
- Clear the caches

Check for requirements

The eZ Components and PHP requirements

The minimum version required of eZ Components with eZ Publish 4.5 is "ezcomponents-ezp45", containing a fix to the package. This version is bundled in eZ Publish 4.5 Enterprise. eZ Publish 4.5 is compatible with PHP version 5.2.1 and above, but certified on RHEL 6 and Debian 6, PHP 5.3.x distros. See <http://ez.no/ezpublish/requirements> for more info.

Step 1: Upgrade the distribution files

The easiest way to upgrade the distribution files is to unpack eZ Publish 4.5 to a separate directory and then copy the directories that contain site-specific files from the existing installation. Make sure that you copy the following directories:

- design/example
- design/example_admin
- var
- settings/siteaccess
- settings/override
- (index_cluster.php)

Cluster note: If you are using cluster mode its necessary to copy index_cluster.php from the old version.

Replace "example" and "example_admin" with the actual names of your site accesses.

Important note: Because the new directory has replaced the original directory, the directory permissions need to be fixed. Use the following commands to do this.

(You have the choice between Shell commands or Alternative shell commands):

- **Shell commands**

These shell commands will give proper permission to the web server:

```
cd </path/to/your/eZ/Publish/directory>
chmod -R a+rw design extension settings var
```

- **Alternative shell commands**

These commands will setup the permission more correctly, but require knowledge about the running web server (change 'nouser' to http server user).

```
cd </path/to/your/eZ/Publish/directory>
chmod -R og+rw design extension settings var
chown -R nouser:nouser design extension settings var
```

Step 2: Upgrade custom extensions

If you are using custom extensions, the sub-directories inside the "extension" directory will also have to be copied. However, make sure that you do not overwrite any extensions that are included in eZ Publish, which currently are eZ Flow (2.x), eZ Online Editor (5.x), eZ OpenOffice Document format (2.x), eZ JSCore (1.x), eZ Image Editor (1.x), eZ Comments (1.x), eZ Multiupload (1.x), eZ MB Password Expiry (1.x), eZ Network (1.x), eZ REST API Provider (1.x), eZ Script Monitor (1.x), eZ SI, eZ Find (2.x). Note that upgrading the distribution files will overwrite the autoload arrays for extensions. You will need to re-generate the autoload arrays for active extensions later.

See the dedicated upgrade instructions for eZ Flow and Website Interface below.

The updated versions of eZ Flow and Website Interface will also install the following extensions:

- eZ Website Toolbar
- eZ Star Rating
- eZ Google Maps Location

Note: For eZ Online Editor 5.x and eZ JS Core you will need to replace the following rewrite rules when using Virtual Hosts:

```
RewriteRule ^/var/cache/texttoimage/..* - [L]
RewriteRule ^/var/[^/]+/cache/(texttoimage|public)/.* - [L]
```

with:

```
RewriteRule ^/var/([~/]+)?cache/(texttoimage|public)/.* - [L]
```

For more detailed instructions, see the dedicated [eZ OE](#) and eZ JS Core doc pages.

Step 3: updateimagesystem.php

If the installation is a new installation or based on any eZ Publish version since 3.3 then you can skip running the "updateimagesystem.php" script. If not, then execute it using:

```
php update/common/scripts/4.1/updateimagesystem.php -s $SITE_ACCESS
```

Step 4: Upgrade the database

The update script for the database is located in the following locations, skip the ones that don't apply to the version your upgrading from:

```
<eZ Publish root>/update/database/<mysql|postgresql>/4.4/
dbupdate-4.3.0-to-4.4.0.sql
<eZ Publish root>/update/database/<mysql|postgresql>/4.5/
dbupdate-4.4.0-to-4.5.0.sql
```

You can run these with the appropriate sql command line tool or application.

MySQL

The "ezmysql" database driver has been deprecated in 4.5 and it is recommended to use the "ezmysqli" driver instead. This requires that the "mysqli" extension in PHP is enabled and can be archived by changing the driver in your override or siteaccesses "site.ini.append.php" settings to:

```
[DatabaseSettings]
DatabaseImplementation=ezmysqli
```

Cluster

The "Mysql DB/DFS Cluster" database back-end has not yet been deprecated, but it is recommended to use the same driver as the main database driver, so change your "file.ini.append.php" settings in override or siteaccesses to use "eZDFSFileHandlerMySQLiBackend" for DFS Cluster or "eZDBFileHandlerMySQLiBackend" for DB Cluster.

Note: The "index_cluster.php" file should be updated to

```
define( 'STORAGE_BACKEND', 'dfsmysqli' )
```

If you use DB cluster, remove "dfs" from the string.

Step 5: Regenerate the autoload array for extensions

The autoload system also has some changes, for example the autoload array for extensions is now placed in var/autoload of your eZ Publish installation (along the class changes in extensions itself).

To regenerate the autoload array, execute the following script from the root of your eZ Publish directory

```
php bin/php/ezpgenerateautoloads.php --extension
```

Step 6: Run the system upgrade scripts

4.3 to 4.4

Update script for this version contained a bug and is repeated in 4.4 to 4.5 bellow, thus is not needed in this instruction.

4.4 to 4.5

For eZ Publish 4.4 which was upgraded from 4.3, run the following command from the root of your eZ Publish directory:

```
php update/common/scripts/4.5/updatesectionidentifier.php -s $SITE_ACCESS
```

Note: Skip this step if this your 4.4 installation is not an upgrade from 4.3. If you are uncertain, please run the script.

Step 7: Enable ezjscore

Both the "admin2" design, "ezoe", "ezwebin", "ezflow" and several other extensions requires the ezjscore extension to be enabled, this is archived by editing your "settings/override/site.ini.append.php" settings file to make sure it looks like this:

```
<pre>
[ExtensionSettings]
ActiveExtensions []
ActiveExtensions []=ezjscore
</pre>
```

Step 8: Enable admin2 design for back-end

To enable admin2 open "site.ini" for your admin site access. For an eZ Flow site this would for instance be:

```
settings/siteaccess/ezflow_site_admin/site.ini.append.php
```

In this file include "AdditionalSiteDesignList[]=admin2" to your [DesignSettings]. It is important that you add the *admin2* design before the *admin* design, in order to let extensions that have admin templates to continue to work. On an eZ Flow site, it would mean changing the settings from:

```
[DesignSettings]
SiteDesign=ezflow_site_admin
AdditionalSiteDesignList []=admin
AdditionalSiteDesignList []=ezflow
```

to

```
[DesignSettings]
SiteDesign=ezflow_site_admin
AdditionalSiteDesignList []=admin2
AdditionalSiteDesignList []=admin
AdditionalSiteDesignList []=ezflow
```

When using a plain or eZ Webin install you should make the same changes. In this case you will of course have a different SiteDesign setting and no AdditionalSiteDesignList[]=ezflow line.

To enable users to change their Administration Interface preferences (Location and Re-Edit editing preferences), add the following line to the [Toolbar_admin_right] block in settings/siteaccess/<siteaccess_name>/toolbar.ini.append.php:

```
Tool []=admin_preferences
```

So the block will then look like like this:

```
[Toolbar_admin_right]
Tool []
Tool []=admin_current_user
Tool []=admin_bookmarks
Tool []=admin_preferences
```

Step 9: Clear the caches

Whenever an eZ Publish solution is upgraded, all caches must be cleared in a proper way. This should be done from within a system shell:

1. Navigate into the eZ Publish 4.4 directory.
2. Run the script using the following shell command:

```
php bin/php/ezcache.php --clear-all --purge
```

Purging ensures that the caches are physically removed. When the "--purge" parameter is not specified, the caches will be expired but not removed.

Sometimes the script is unable to clear all cache files because of restrictive file/directory permission settings. Make sure that all cache files have been cleared by inspecting the contents of the various cache sub-directories within the "var" directory (typically the "var/cache/" and "var/<name_of_siteaccess>/cache/" directories). If there are any cache files left, you need to remove them manually.

Step 10: Upgrade extensions

When using eZ Webin and eZ Flow, these extension will also need to be updated.

The first thing to do is to change the extensions permissions. An example on how this is done:

```
$chmod -R a+rwX extension
```

4.3 to 4.4:

eZ Webin 1.6 to 1.7

or

eZ Flow 2.1 to 2.2

4.4 to 4.5:

eZ Webin 1.7 to 1.8

or

eZ Flow 2.2 to 2.3

Share your information

1.4.2 Direct upgrading to 4.6 from 4.1, 4.2, 4.3 and 4.4

Please note that there are various documentation options for the direct upgrading to eZ Publish 4.6. There are 4 separate upgrade kits containing the information for upgrading from versions 4.1, 4.2, 4.3 and 4.4 to 4.6 of eZ Publish.

- Direct upgrading from 4.1 to 4.6 (page [43](#))
- Direct upgrading from 4.2 to 4.6 (page [53](#))
- Direct upgrading from 4.3 to 4.6 (page [62](#))
- Direct upgrading from 4.4 to 4.6 (page [70](#))

Direct upgrading from 4.1 to 4.6

This section describes how to upgrade your existing eZ Publish 4.1 installation directly to version 4.6. Make sure that you have a working backup of the site before you do the actual upgrade, and make sure the installation you are performing the upgrade on is offline.

Important upgrade notes:

The procedure for upgrading directly from version 4.1.x, 4.2, 4.3 or 4.4 to 4.6 consists of the following steps:

- Upgrade the distribution files to 4.6
- Upgrade custom extensions
 - 4.1.0 to 4.2
 - 4.2 to 4.3
 - 4.3 to 4.4
 - 4.4 to 4.5
 - 4.5 to 4.6
- Upgrade the database to 4.6
 - 4.1.0 to 4.2
 - 4.2 to 4.3
 - 4.3 to 4.4
 - 4.4 to 4.5
 - 4.5 to 4.6
- Regenerate the autoload array for extensions
- Run the system upgrade scripts for 4.6
 - 4.1.0 to 4.2
 - 4.2 to 4.3
 - 4.3 to 4.4
 - 4.4 to 4.5
 - 4.5 to 4.6
- Additional upgrade step
- Enable ezjscore
- Enable admin2-design for back-end
- Clear the caches
- Upgrade extensions

Check for requirements

The eZ Components and PHP requirements

The minimum version required of eZ Components with eZ Publish 4.6 is "ezcomponents-ezp46", containing a fix to the package. This version is bundled in eZ Publish 4.6 Enterprise. eZ Publish 4.6 is compatible with PHP version 5.2.1 and above, but certified on RHEL 6 and Debian 6, PHP 5.3.x distros. See <http://ez.no/ezpublish/requirements> for more info.

Step 1: Upgrade the distribution files

The easiest way to upgrade the distribution files is to unpack eZ Publish 4.6 to a separate directory and then copy the directories that contain site-specific files from the existing installation. Make sure that you copy the following directories:

- design/example
- design/example_admin
- var
- settings/siteaccess
- settings/override
- (index_cluster.php)

Cluster note: If you are using cluster mode its necessary to copy index_cluster.php from the old version.

Replace "example" and "example_admin" with the actual names of your site accesses.

Important note: Because the new directory has replaced the original directory, the directory permissions need to be fixed. Use the following commands to do this.

(You have the choice between Shell commands or Alternative shell commands):

- **Shell commands**

These shell commands will give proper permission to the web server:

```
cd </path/to/your/eZ/Publish/directory>
chmod -R a+rw design extension settings var
```

- **Alternative shell commands**

These commands will setup the permission more correctly, but require knowledge about the running web server (change 'nouser' to http server user).

```
cd </path/to/your/eZ/Publish/directory>
chmod -R og+rwx design extension settings var
chown -R nouser:nouser design extension settings var
```

Step 2: Custom extensions

If you are using custom extensions, the sub-directories inside the "extension" directory will also have to be copied. However, make sure that you do not overwrite any extensions that are included in eZ Publish, which currently are eZ Flow (2.x), eZ Online Editor (5.x), eZ OpenOffice Document format (2.x), eZ JSCore (1.x), eZ Image Editor (1.x), eZ Comments (1.x), eZ Multiupload (1.x), eZ MB Password Expiry (1.x), eZ Network (1.x), eZ REST API Provider (1.x), eZ Script Monitor (1.x), eZ SI, eZ Find (2.x). Note that upgrading the distribution files will overwrite the autoload arrays for extensions. You will need to re-generate the autoload arrays for active extensions later.

See the dedicated upgrade instructions for eZ Flow and Website Interface below.

The updated versions of eZ Flow and Website Interface will also install the following extensions:

- eZ Website Toolbar
- eZ Star Rating
- eZ Google Maps Location

Note: For eZ Online Editor 5.x and eZ JS Core you will need to replace the following rewrite rules when using Virtual Hosts:

```
RewriteRule ^/var/cache/texttoimage/./* - [L]
RewriteRule ^/var/[^/]+/cache/(texttoimage|public)/.* - [L]
```

with:

```
RewriteRule ^/var/([^/]+)?cache/(texttoimage|public)/.* - [L]
```

For more detailed instructions, see the dedicated [eZ OE](#) and [eZ JS Core](#) doc pages.

Step 3: updateimagesystem.php

This script is necessary to run if you have data of the "ezimage" datatype dating from before eZ Publish 3.3. In version 3.3 a new image system took over based on "ezimagealiasandler".

If you are upgrading to the 4.2 series of eZ Publish for the first time, and the installation has been running since prior to eZ Publish 3.3 then you need to run the "updateimagesystem.php" script before running any of the "dbupdate" scripts.

If the installation is a new installation or based on any eZ Publish version since 3.3 then you can skip running the "updateimagesystem.php" script.

If not, then execute it using:

```
php update/common/scripts/4.1/updateimagesystem.php -s $SITE_ACCESS
```

Step 4: Upgrade the database

The update script for the database is located in the following locations, skip the ones that don't apply to the version your upgrading from:

```
<eZ Publish root>/update/database/<mysql|postgresql>/4.2/  
dbupdate-4.1.0-to-4.2.0.sql  
<eZ Publish root>/update/database/<mysql|postgresql>/4.3/  
dbupdate-4.2.0-to-4.3.0.sql  
<eZ Publish root>/update/database/<mysql|postgresql>/4.4/  
dbupdate-4.3.0-to-4.4.0.sql  
<eZ Publish root>/update/database/<mysql|postgresql>/4.5/  
dbupdate-4.4.0-to-4.5.0.sql  
<eZ Publish root>/update/database/<mysql|postgresql>/4.6/  
dbupdate-4.5.0-to-4.6.0.sql
```

You can run these with the appropriate sql command line tool or application.

Note on 4.1.x to 4.2 upgrade: You need to selectively choose the correct sql from first file above if your upgrading from 4.1.1, 4.1.2, 4.1.3 or 4.1.4.

If your upgrading from 4.1.1: Ignore everything above -- END: from 4.1.1.

If your upgrading from 4.1.2 or 4.1.3: Ignore everything above -- END: from 4.1.2.

If your upgrading from 4.1.4: Ignore everything above -- END: from 4.1.4.

Clustering:

If your upgrading from 4.1 or 4.2 using db or dfs cluster, make sure you apply the following sql updates (but for dfs make sure you change table name from "ezdbfile" to "ezdfsfile"):

```
<eZ Publish root>/update/database/mysql/4.3/dbupdate-cluster-4.2.0-to-4.3.0.sql
```

Schema changes

Note: These schema changes are only necessary to do if you have eZ Flow installed and if you are making the following upgrades:

4.1.1 -> 4.6

4.1.2 -> 4.6

4.1.3 -> 4.6
4.1.4 -> 4.6
4.2 -> 4.6

You must run this sql on the mysql client

```
ALTER TABLE ezm_pool ADD INDEX ezm_pool__block_id__ts_hidden ( block_id,  
ts_hidden );
```

MySQL

The "ezmysql" database driver has been deprecated in 4.5 and it is recommended to use the "ezmysqli" driver instead. This requires that the "mysqli" extension in PHP is enabled and can be archived by changing the driver in your override or siteaccesses "site.ini.append.php" settings to:

```
[DatabaseSettings]  
DatabaseImplementation=ezmysqli
```

Cluster

The "Mysql DB/DFS Cluster" database back-end has not yet been deprecated, but it is recommended to use the same driver as the main database driver, so change your "file.ini.append.php" settings in override or siteaccesses to use "eZDFSFileHandlerMySQLiBackend" for DFS Cluster or "eZDBFileHandlerMySQLiBackend" for DB Cluster.

Note: The "index_cluster.php" file should be updated to

```
define( 'STORAGE_BACKEND', 'dfsmysqli' )
```

If you use DB cluster, remove "dfs" from the string.

Step 5: Regenerate the autoload array for extensions

The autoload system also has some changes, for example the autoload array for extensions is now placed in var/autoload of your eZ Publish installation (along the class changes in extensions itself).

To regenerate the autoload array, execute the following script from the root of your eZ Publish directory:

```
php bin/php/ezpgenerateautoloads.php --extension
```

Step 6: Run the system upgrade scripts

4.1 to 4.2

- `correctxmlalign.php`: after upgrading to 4.1 the default alignment for embedded images in the "ezxmltext" datatype is removed, this script will allow you to have the correct xmlalignment.

```
php update/common/scripts/4.1/correctxmlalign.php -s $SITE_ACCESS
```

- `fixorphanimages.php`: this script will delete images from the storage directory that are no longer connected to any content object attribute

```
php update/common/scripts/4.2/fixorphanimages.php -s $SITE_ACCESS
```

4.2 to 4.3

This version includes a script to fix #15478: Node assignment is not removed when removing node from child list:

```
php update/common/scripts/4.3/updatenodeassignment.php -s $SITE_ACCESS
```

4.3 to 4.4

Update script for this version contained a bug and is repeated in 4.4 to 4.5 below, thus is not needed in this instruction.

4.4 to 4.5

For eZ Publish 4.4 which was upgraded from 4.3, run the following command from the root of your eZ Publish directory:

```
php update/common/scripts/4.5/updatesectionidentifier.php -s $SITE_ACCESS
```

Note: Skip this step if this your 4.4 installation is not an upgrade from 4.3. If you are uncertain, please run the script.

4.5 to 4.6

There are two system upgrade scripts that can be run for the eZ Publish 4.6 upgrade, "`remove-trashedimages.php`" and "`updateordernumber.php`".

Script 1: If you want to ensure no trashed images remains on your setup because of image aliases not being restored when restoring object from trash (issue #017781), you do the following:

1. make sure that no items remains in trash for any users (images from objects in trash would be impossible to restore not doing so)
2. run the following cleanup script by executing this command:

```
$ php update/common/scripts/4.6/removetrashedimages.php
```

Note: It is safe to skip this test, this is only to free extra space.

Script 2: You need to run an upgrade script to update webshop order numbers issue (#018233) by running:

```
$ php update/common/scripts/4.6/updateordernumber.php
```

Note: If you dont have webshop or you dont have orders (table "ezorder") in your eZ Publish, you dont have to run this script.

Step 7: Additional upgrade steps

Update "override.ini.append.php"

Content panes are not visible on the "User accounts" tab in the back-end. To fix it on an already-installed eZ Publish, remove the following rules from your "override.ini.append.php" for your admin siteaccess:

```
[window_controls]
Source=window_controls.tpl
MatchFile=window_controls_user.tpl
Subdir=templates
Match[navigation_part_identifier]=ezusernavigationpart

[windows]
Source=windows.tpl
MatchFile=windows_user.tpl
Subdir=templates
Match[navigation_part_identifier]=ezusernavigationpart
```

... and clear the cache after removing. See separate step below.

Step 8: Enable ezjscore

Both the "admin2" design, "ezoe", "ezwebin", "ezflow" and several other extensions requires the ezjscore extension to be enabled, this is archived by editing your "settings/override/site.ini.append.php" settings file to make sure it looks like this:

```
<pre>
[ExtensionSettings]
ActiveExtensions []
ActiveExtensions []=ezjscore
</pre>
```

Step 9: Enable admin2 design for back-end

To enable admin2 open "site.ini" for your admin site access. For an eZ Flow site this would for instance be:

```
settings/siteaccess/ezflow_site_admin/site.ini.append.php
```

In this file include "AdditionalSiteDesignList[]=admin2" to your [DesignSettings]. It is important that you add the *admin2* design before the *admin* design, in order to let extensions that have admin templates to continue to work. On an eZ Flow site, it would mean changing the settings from:

```
[DesignSettings]
SiteDesign=ezflow_site_admin
AdditionalSiteDesignList []=admin
AdditionalSiteDesignList []=ezflow
```

to

```
[DesignSettings]
SiteDesign=ezflow_site_admin
AdditionalSiteDesignList []=admin2
AdditionalSiteDesignList []=admin
AdditionalSiteDesignList []=ezflow
```

When using a plain or eZ Webin install you should make the same changes. In this case you will of course have a different SiteDesign setting and no AdditionalSiteDesignList[]=ezflow line.

To enable users to change their Administration Interface preferences (Location and Re-Edit editing preferences), add the following line to the [Toolbar_admin_right] block in settings/siteaccess/<siteaccess_name>/toolbar.ini.append.php:

```
Tool []=admin_preferences
```

So the block will then look like like this:

```
[Toolbar_admin_right]
Tool []
Tool []=admin_current_user
Tool []=admin_bookmarks
Tool []=admin_preferences
```

Step 10: Clear the caches

Whenever an eZ Publish solution is upgraded, all caches must be cleared in a proper way. **Note:** You should not be a root user when you start the caching process (refer to <http://issues.ez.no/15823>). The caching should be done from within a system shell:

1. Navigate into the eZ Publish 4.5 directory.
2. Run the script using the following shell command:

```
php bin/php/ezcache.php --clear-all --purge
```

Purging ensures that the caches are physically removed. When the "--purge" parameter is not specified, the caches will be expired but not removed.

Sometimes the script is unable to clear all cache files because of restrictive file/directory permission settings. Make sure that all cache files have been cleared by inspecting the contents of the various cache sub-directories within the "var" directory (typically the "var/cache/" and "var/<name_of_siteaccess>/cache/" directories). If there are any cache files left, you need to remove them manually.

Step 11: Upgrade extensions

When using eZ Webin and eZ Flow, these extension will also need to be updated.

The first thing to do is to change the extensions permissions. An example on how this is done:

```
$chmod -R a+rx extension
```

4.1 to 4.2:

eZ Webin 1.4 to 1.5
or
eZ Flow 1.1 to 2.0

4.2 to 4.3:

eZ Webin 1.5 to 1.6

or
eZ Flow 2.0 to 2.1

4.3 to 4.4:

eZ Webin 1.6 to 1.7
or
eZ Flow 2.1 to 2.2

4.4 to 4.5:

eZ Webin 1.7 to 1.8
or
eZ Flow 2.2 to 2.3

4.5 to 4.6:

eZ Webin 1.8 to 1.9
or
eZ Flow 2.3 to 2.4

5
1
Year
information

1

Direct upgrading from 4.2 to 4.6

This section describes how to upgrade your existing eZ Publish 4.2 installation directly to version 4.6. Make sure that you have a working backup of the site before you do the actual upgrade, and make sure the installation you are performing the upgrade on is offline.

Important upgrade notes:

The procedure for upgrading directly from version 4.2 to 4.6 consists of the following steps:

- Upgrade the distribution files to 4.6
- Upgrade custom extensions
 - 4.2 to 4.3
 - 4.3 to 4.4
 - 4.4 to 4.5
 - 4.5 to 4.6
- Upgrade the database to 4.6
 - 4.2 to 4.3
 - 4.3 to 4.4
 - 4.4 to 4.5
 - 4.5 to 4.6
- Regenerate the autoload array for extensions
- Run the system upgrade scripts for 4.6
 - 4.2 to 4.3
 - 4.3 to 4.4
 - 4.4 to 4.5
 - 4.5 to 4.6
- Additional upgrade step
- Enable ezjscore
- Enable admin2-design for back-end
- Clear the caches
- Upgrade extensions

Check for requirements

The eZ Components and PHP requirements

The minimum version required of eZ Components with eZ Publish 4.6 is "ezcomponents-ezp46", containing a fix to the package. This version is bundled in eZ Publish 4.6 Enterprise. eZ Publish 4.6 is compatible with PHP version 5.2.1 and above, but certified on RHEL 6 and Debian 6, PHP 5.3.x distros. See <http://ez.no/ezpublish/requirements> for more info.

Step 1: Upgrade the distribution files

The easiest way to upgrade the distribution files is to unpack eZ Publish 4.6 to a separate directory and then copy the directories that contain site-specific files from the existing installation. Make sure that you copy the following directories:

- design/example
- design/example_admin
- var
- settings/siteaccess
- settings/override
- (index_cluster.php)

Cluster note: If you are using cluster mode its necessary to copy index_cluster.php from the old version.

Replace "example" and "example_admin" with the actual names of your site accesses.

Important note: Because the new directory has replaced the original directory, the directory permissions need to be fixed. Use the following commands to do this.

(You have the choice between Shell commands or Alternative shell commands):

- **Shell commands**

These shell commands will give proper permission to the web server:

```
cd </path/to/your/eZ/Publish/directory>
chmod -R a+rw design extension settings var
```

- **Alternative shell commands**

These commands will setup the permission more correctly, but require knowledge about the running web server (change 'nouser' to http server user).

```
cd </path/to/your/eZ/Publish/directory>
chmod -R og+rx design extension settings var
chown -R nouser:nouser design extension settings var
```

Step 2: Custom extensions

If you are using custom extensions, the sub-directories inside the "extension" directory will also have to be copied. However, make sure that you do not overwrite any extensions that are included in eZ Publish, which currently are eZ Flow (2.x), eZ Online Editor (5.x), eZ OpenOffice Document format (2.x), eZ JSCore (1.x), eZ Image Editor (1.x), eZ Comments (1.x), eZ Multiupload (1.x), eZ MB Password Expiry (1.x), eZ Network (1.x), eZ REST API Provider (1.x), eZ Script Monitor (1.x), eZ SI, eZ Find (2.x). Note that upgrading the distribution files will overwrite the autoload arrays for extensions. You will need to re-generate the autoload arrays for active extensions later.

See the dedicated upgrade instructions for eZ Flow and Website Interface below.

The updated versions of eZ Flow and Website Interface will also install the following extensions:

- eZ Website Toolbar
- eZ Star Rating
- eZ Google Maps Location

Note: For eZ Online Editor 5.x and eZ JS Core you will need to replace the following rewrite rules when using Virtual Hosts:

```
RewriteRule ^/var/cache/texttoimage/./* - [L]
RewriteRule ^/var/[^/]+/cache/(texttoimage|public)/.* - [L]
```

with:

```
RewriteRule ^/var/([~/]+)?cache/(texttoimage|public)/.* - [L]
```

For more detailed instructions, see the dedicated [eZ OE](#) and eZ JS Core doc pages.

Step 3: updateimagesystem.php

If the installation is a new installation or based on any eZ Publish version since 3.3 then you can skip running the "updateimagesystem.php" script. If not, then execute it using:

```
php update/common/scripts/4.1/updateimagesystem.php -s $SITE_ACCESS
```

Step 4: Upgrade the database

The update script for the database is located in the following locations, skip the ones that don't apply to the version your upgrading from:

```
<eZ Publish root>/update/database/<mysql|postgresql>/4.3/  
dbupdate-4.2.0-to-4.3.0.sql  
<eZ Publish root>/update/database/<mysql|postgresql>/4.4/  
dbupdate-4.3.0-to-4.4.0.sql  
<eZ Publish root>/update/database/<mysql|postgresql>/4.5/  
dbupdate-4.4.0-to-4.5.0.sql  
<eZ Publish root>/update/database/<mysql|postgresql>/4.6/  
dbupdate-4.4.0-to-4.5.0.sql
```

You can run these with the appropriate sql command line tool or application.

Clustering:

If your upgrading from 4.1 or 4.2 using db or dfs cluster, make sure you apply the following sql updates (but for dfs make sure you change table name from "ezdbfile" to "ezdfsfile"):

```
<eZ Publish root>/update/database/mysql/4.3/dbupdate-cluster-4.2.0-to-4.3.0.sql
```

Schema changes

Note: The schema changes are only necessary if you have eZ Flow installed and if you are making an upgrade from 4.2 -> 4.6. If you must run this sql on the mysql client

```
ALTER TABLE ezm_pool ADD INDEX ezm_pool__block_id__ts_hidden ( block_id,  
ts_hidden );
```

MySQL

The "ezmysql" database driver has been deprecated in 4.5 and it is recommended to use the "ezmysqli" driver instead. This requires that the "mysqli" extension in PHP is enabled and can be archived by changing the driver in your override or siteaccesses "site.ini.append.php" settings to:

```
[DatabaseSettings]
DatabaseImplementation=eZmysqli
```

Cluster

The "Mysql DB/DFS Cluster" database back-end has not yet been deprecated, but it is recommended to use the same driver as the main database driver, so change your "file.ini.append.php" settings in override or siteaccesses to use "eZDFSFileHandlerMySQLiBackend" for DFS Cluster or "eZDBFileHandlerMysqliBackend" for DB Cluster.

Note: The "index_cluster.php" file should be updated to

```
define( 'STORAGE_BACKEND', 'dfsmysqli' )
```

If you use DB cluster, remove "dfs" from the string.

Step 5: Regenerate the autoload array for extensions

The autoload system also has some changes, for example the autoload array for extensions is now placed in var/autoload of your eZ Publish installation (along the class changes in extensions itself).

To regenerate the autoload array, execute the following script from the root of your eZ Publish directory:

```
php bin/php/ezpgenerateautoloads.php --extension
```

Step 6: Run the system upgrade scripts

4.2 to 4.3

This version includes a script to fix #15478: Node assignment is not removed when removing node from child list:

```
php update/common/scripts/4.3/updatenodeassignment.php -s $SITE_ACCESS
```

4.3 to 4.4

Update script for this version contained a bug and is repeated in 4.4 to 4.5 below, thus is not needed in this instruction.

4.4 to 4.5

For eZ Publish 4.4 which was upgraded from 4.3, run the following command from the root of your eZ Publish directory:

```
php update/common/scripts/4.5/updatesectionidentifier.php -s $SITE_ACCESS
```

Note: Skip this step if this your 4.4 installation is not an upgrade from 4.3. If you are uncertain, please run the script.

4.5 to 4.6

There are two system upgrade scripts that can be run for the eZ Publish 4.6 upgrade, "*remove-trashedimages.php*" and "*updateordernumber.php*".

Script 1: If you want to ensure no trashed images remains on your setup because of image aliases not being restored when restoring object from trash (issue #017781), you do the following:

1. make sure that no items remains in trash for any users (images from objects in trash would be impossible to restore not doing so)
2. run the following cleanup script by executing this command:

```
$ php update/common/scripts/4.6/removetrashedimages.php
```

Note: It is safe to skip this test, this is only to free extra space.

Script 2: You need to run an upgrade script to update webshop order numbers issue (#018233) by running:

```
$ php update/common/scripts/4.6/updateordernumber.php
```

Note: If you dont have webshop or you dont have orders (table "*ezorder*") in your eZ Publish, you dont have to run this script.

Step 7: Additional upgrade steps

Update "*override.ini.append.php*"

Content panes are not visible on the "User accounts" tab in the back-end. To fix it on an already-installed eZ Publish, remove the following rules from your "*override.ini.append.php*" for your admin siteaccess:

```
[window_controls]
Source=window_controls.tpl
MatchFile=window_controls_user.tpl
Subdir=templates
Match[navigation_part_identifier]=ezusernavigationpart

[windows]
Source=windows.tpl
MatchFile=windows_user.tpl
Subdir=templates
Match[navigation_part_identifier]=ezusernavigationpart
```

... and clear the cache after removing. See separate step below.

Step 8: Enable ezjscore

Both the "admin2" design, "ezoe", "ezwebin", "ezflow" and several other extensions requires the ezjscore extension to be enabled, this is archived by editing your "settings/override/site.ini.append.php" settings file to make sure it looks like this:

```
<pre>
[ExtensionSettings]
ActiveExtensions []
ActiveExtensions []=ezjscore
</pre>
```

Step 9: Enable admin2 design for back-end

To enable admin2 open "site.ini" for your admin site access. For an eZ Flow site this would for instance be:

```
settings/siteaccess/ezflow_site_admin/site.ini.append.php
```

In this file include "AdditionalSiteDesignList[]=admin2" to your [DesignSettings]. It is important that you add the *admin2* design before the *admin* design, in order to let extensions that have admin templates to continue to work. On an eZ Flow site, it would mean changing the settings from:

```
[DesignSettings]
SiteDesign=ezflow_site_admin
AdditionalSiteDesignList []=admin
AdditionalSiteDesignList []=ezflow
```

to

```
[DesignSettings]
SiteDesign=ezflow_site_admin
AdditionalSiteDesignList []=admin2
AdditionalSiteDesignList []=admin
AdditionalSiteDesignList []=ezflow
```

When using a plain or eZ Webin install you should make the same changes. In this case you will of course have a different SiteDesign setting and no AdditionalSiteDesignList []=ezflow line.

To enable users to change their Administration Interface preferences (Location and Re-Edit editing preferences), add the following line to the [Toolbar_admin_right] block in settings/siteaccess/<siteaccess_name>/toolbar.ini.append.php:

```
Tool []=admin_preferences
```

So the block will then look like like this:

```
[Toolbar_admin_right]
Tool []
Tool []=admin_current_user
Tool []=admin_bookmarks
Tool []=admin_preferences
```

Step 10: Clear the caches

Whenever an eZ Publish solution is upgraded, all caches must be cleared in a proper way. **Note:** You should not be a root user when you start the caching process (refer to <http://issues.ez.no/15823>). The caching should be done from within a system shell:

1. Navigate into the eZ Publish 4.5 directory.
2. Run the script using the following shell command:

```
php bin/php/ezcache.php --clear-all --purge
```

Purging ensures that the caches are physically removed. When the "--purge" parameter is not specified, the caches will be expired but not removed.

Sometimes the script is unable to clear all cache files because of restrictive file/directory permission settings. Make sure that all cache files have been cleared by inspecting the contents of the various cache sub-directories within the "var" directory (typically the "var/cache/" and "var/<name_of_siteaccess>/cache/" directories). If there are any cache files left, you need to remove them manually.

Step 11: Upgrade extensions

When using eZ Webin and eZ Flow, these extension will also need to be updated.

The first thing to do is to change the extensions permissions. An example on how this is done:

```
$chmod -R a+rwX extension
```

4.2 to 4.3:

eZ Webin 1.5 to 1.6
or
eZ Flow 2.0 to 2.1

4.3 to 4.4:

eZ Webin 1.6 to 1.7
or
eZ Flow 2.1 to 2.2

4.4 to 4.5:

eZ Webin 1.7 to 1.8
or
eZ Flow 2.2 to 2.3

4.5 to 4.6:

eZ Webin 1.8 to 1.9
or
eZ Flow 2.3 to 2.4

Direct upgrading from 4.3 to 4.6

This section describes how to upgrade your existing eZ Publish 4.3 installation directly to version 4.6. Make sure that you have a working backup of the site before you do the actual upgrade, and make sure the installation you are performing the upgrade on is offline.

Important upgrade notes:

The procedure for upgrading directly from version 4.3 to 4.6 consists of the following steps:

- Upgrade the distribution files to 4.6
- Upgrade custom extensions
 - 4.3.to 4.4
 - 4.4 to 4.5
 - 4.5 to 4.6
- Upgrade the database to 4.6
 - 4.3 to 4.4
 - 4.4 to 4.5
 - 4.5 to 4.6
- Regenerate the autoload array for extensions
- Run the system upgrade scripts for 4.6
 - 4.3 to 4.4
 - 4.4 to 4.5
 - 4.5 to 4.6
- Additional upgrade step
- Enable ezjscore
- Enable admin2-design for back-end
- Clear the caches
- Upgrade extensions

Check for requirements

The eZ Components and PHP requirements

The minimum version required of eZ Components with eZ Publish 4.6 is "ezcomponents-ezp46", containing a fix to the package. This version is bundled in eZ Publish 4.6 Enterprise. eZ Publish 4.6 is compatible with PHP version 5.2.1 and above, but certified on RHEL 6 and Debian 6, PHP 5.3.x distros. See <http://ez.no/ezpublish/requirements> for more info.

Step 1: Upgrade the distribution files

The easiest way to upgrade the distribution files is to unpack eZ Publish 4.6 to a separate directory and then copy the directories that contain site-specific files from the existing installation. Make sure that you copy the following directories:

- design/example
- design/example_admin
- var
- settings/siteaccess
- settings/override
- (index_cluster.php)

Cluster note: If you are using cluster mode its necessary to copy index_cluster.php from the old version.

Replace "example" and "example_admin" with the actual names of your site accesses.

Important note: Because the new directory has replaced the original directory, the directory permissions need to be fixed. Use the following commands to do this.

(You have the choice between Shell commands or Alternative shell commands):

- **Shell commands**

These shell commands will give proper permission to the web server:

```
cd </path/to/your/eZ/Publish/directory>
chmod -R a+rw design extension settings var
```

- **Alternative shell commands**

These commands will setup the permission more correctly, but require knowledge about the running web server (change 'nouser' to http server user).

```
cd </path/to/your/eZ/Publish/directory>
chmod -R og+rw design extension settings var
chown -R nouser:nouser design extension settings var
```

Step 2: Upgrade custom extensions

If you are using custom extensions, the sub-directories inside the "extension" directory will also have to be copied. However, make sure that you do not overwrite any extensions that are included in eZ Publish, which currently are eZ Flow (2.x), eZ Online Editor (5.x), eZ OpenOffice Document format (2.x), eZ JSCore (1.x), eZ Image Editor (1.x), eZ Comments (1.x), eZ Multiupload (1.x), eZ MB Password Expiry (1.x), eZ Network (1.x), eZ REST API Provider (1.x), eZ Script Monitor (1.x), eZ SI, eZ Find (2.x). Note that upgrading the distribution files will overwrite the autoload arrays for extensions. You will need to re-generate the autoload arrays for active extensions later.

See the dedicated upgrade instructions for eZ Flow and Website Interface below.

The updated versions of eZ Flow and Website Interface will also install the following extensions:

- eZ Website Toolbar
- eZ Star Rating
- eZ Google Maps Location

Note: For eZ Online Editor 5.x and eZ JS Core you will need to replace the following rewrite rules when using Virtual Hosts:

```
RewriteRule ^/var/cache/texttoimage/./* - [L]
RewriteRule ^/var/[~/]+/cache/(texttoimage|public)/.* - [L]
```

with:

```
RewriteRule ^/var/([~/]+)?cache/(texttoimage|public)/.* - [L]
```

For more detailed instructions, see the dedicated [eZ OE](#) and [eZ JS Core](#) doc pages.

Step 3: updateimagesystem.php

If the installation is a new installation or based on any eZ Publish version since 3.3 then you can skip running the "updateimagesystem.php" script. If not, then execute it using:

```
php update/common/scripts/4.1/updateimagesystem.php -s $SITE_ACCESS
```

Step 4: Upgrade the database

The update script for the database is located in the following locations, skip the ones that don't apply to the version your upgrading from:

```
<eZ Publish root>/update/database/<mysql|postgresql>/4.4/  
dbupdate-4.3.0-to-4.4.0.sql  
<eZ Publish root>/update/database/<mysql|postgresql>/4.5/  
dbupdate-4.4.0-to-4.5.0.sql  
<eZ Publish root>/update/database/<mysql|postgresql>/4.6/  
dbupdate-4.5.0-to-4.6.0.sql
```

You can run these with the appropriate sql command line tool or application.

MySQL

The "ezmysql" database driver has been deprecated in 4.5 and it is recommended to use the "ezmysql" driver instead. This requires that the "mysqli" extension in PHP is enabled and can be archived by changing the driver in your override or siteaccesses "site.ini.append.php" settings to:

```
[DatabaseSettings]  
DatabaseImplementation=ezmysql
```

Cluster

The "Mysql DB/DFS Cluster" database back-end has not yet been deprecated, but it is recommended to use the same driver as the main database driver, so change your "file.ini.append.php" settings in override or siteaccesses to use "eZDFSFileHandlerMySQLiBackend" for DFS Cluster or "eZDBFileHandlerMySQLiBackend" for DB Cluster.

Note: The "index_cluster.php" file should be updated to

```
define( 'STORAGE_BACKEND', 'dfsmysqli' )
```

If you use DB cluster, remove "dfs" from the string.

Step 5: Regenerate the autoload array for extensions

The autoload system also has some changes, for example the autoload array for extensions is now placed in var/autoload of your eZ Publish installation (along the class changes in extensions itself).

To regenerate the autoload array, execute the following script from the root of your eZ Publish directory:

```
php bin/php/ezpgenerateautoloads.php --extension
```

Step 6: Run the system upgrade scripts

4.3 to 4.4

Update script for this version contained a bug and is repeated in 4.4 to 4.5 below, thus is not needed in this instruction.

4.4 to 4.5

For eZ Publish 4.4 which was upgraded from 4.3, run the following command from the root of your eZ Publish directory:

```
php update/common/scripts/4.5/updatesectionidentifier.php -s $SITE_ACCESS
```

Note: Skip this step if this your 4.4 installation is not an upgrade from 4.3. If you are uncertain, please run the script.

4.5 to 4.6

There are two system upgrade scripts that can be run for the eZ Publish 4.6 upgrade, "*remove-trashedimages.php*" and "*updateordernumber.php*".

Script 1: If you want to ensure no trashed images remains on your setup because of image aliases not being restored when restoring object from trash (issue #017781), you do the following:

1. make sure that no items remains in trash for any users (images from objects in trash would be impossible to restore not doing so)
2. run the following cleanup script by executing this command:

```
$ php update/common/scripts/4.6/removetrashedimages.php
```

Note: It is safe to skip this test, this is only to free extra space.

Script 2: You need to run an upgrade script to update webshop order numbers issue (#018233) by running:

```
$ php update/common/scripts/4.6/updateordernumber.php
```

Note: If you dont have webshop or you dont have orders (table "*ezorder*") in your eZ Publish, you dont have to run this script.

Step 7: Additional upgrade steps

Update "override.ini.append.php"

Content panes are not visible on the "User accounts" tab in the back-end. To fix it on an already-installed eZ Publish, remove the following rules from your "override.ini.append.php" for your admin siteaccess:

```
[window_controls]
Source=window_controls.tpl
MatchFile=window_controls_user.tpl
Subdir=templates
Match[navigation_part_identifier]=ezusernavigationpart

[windows]
Source=windows.tpl
MatchFile=windows_user.tpl
Subdir=templates
Match[navigation_part_identifier]=ezusernavigationpart
```

... and clear the cache after removing. See separate step below.

Step 8: Enable ezjscore

Both the "admin2" design, "ezoe", "ezwebin", "ezflow" and several other extensions requires the ezjscore extension to be enabled, this is archived by editing your "settings/override/site.ini.append.php" settings file to make sure it looks like this:

```
<pre>
[ExtensionSettings]
ActiveExtensions []
ActiveExtensions []=ezjscore
</pre>
```

Step 9: Enable admin2 design for back-end

To enable admin2 open "site.ini" for your admin site access. For an eZ Flow site this would for instance be:

```
settings/siteaccess/ezflow_site_admin/site.ini.append.php
```

In this file include "AdditionalSiteDesignList[]=admin2" to your [DesignSettings]. It is important that you add the *admin2* design before the *admin* design, in order to let extensions that have admin templates to continue to work. On an eZ Flow site, it would mean changing the settings from:

```
[DesignSettings]
SiteDesign=ezflow_site_admin
AdditionalSiteDesignList []=admin
AdditionalSiteDesignList []=ezflow
```

to

```
[DesignSettings]
SiteDesign=ezflow_site_admin
AdditionalSiteDesignList []=admin2
AdditionalSiteDesignList []=admin
AdditionalSiteDesignList []=ezflow
```

When using a plain or eZ Webin install you should make the same changes. In this case you will of course have a different SiteDesign setting and no AdditionalSiteDesignList[]=ezflow line.

To enable users to change their Administration Interface preferences (Location and Re-Edit editing preferences), add the following line to the [Toolbar_admin_right] block in settings/siteaccess/<siteaccess_name>/toolbar.ini.append.php:

```
Tool []=admin_preferences
```

So the block will then look like like this:

```
[Toolbar_admin_right]
Tool []
Tool []=admin_current_user
Tool []=admin_bookmarks
Tool []=admin_preferences
```

Step 10: Clear the caches

Whenever an eZ Publish solution is upgraded, all caches must be cleared in a proper way. This should be done from within a system shell:

1. Navigate into the eZ Publish 4.5 directory.
2. Run the script using the following shell command:

```
php bin/php/ezcache.php --clear-all --purge
```

Purging ensures that the caches are physically removed. When the "--purge" parameter is not specified, the caches will be expired but not removed.

Sometimes the script is unable to clear all cache files because of restrictive file/directory permission settings. Make sure that all cache files have been cleared by inspecting the contents of the various cache sub-directories within the "var" directory (typically the "var/cache/" and "var/<name_of_siteaccess>/cache/" directories). If there are any cache files left, you need to remove them manually.

Step 11: Upgrade extensions

When using eZ Webin and eZ Flow, these extension will also need to be updated.

The first thing to do is to change the extensions permissions. An example on how this is done:

```
$chmod -R a+rx extension
```

4.3 to 4.4:

eZ Webin 1.6 to 1.7
or
eZ Flow 2.1 to 2.2

4.4 to 4.5:

eZ Webin 1.7 to 1.8
or
eZ Flow 2.2 to 2.3

4.5 to 4.6:

eZ Webin 1.8 to 1.9
or
eZ Flow 2.3 to 2.4

Direct upgrading from 4.4 to 4.6

This section describes how to upgrade your existing eZ Publish 4.4 installation directly to version 4.6. Make sure that you have a working backup of the site before you do the actual upgrade, and make sure the installation you are performing the upgrade on is offline.

Important upgrade notes:

The procedure for upgrading directly from version 4.4 to 4.6 consists of the following steps:

- Upgrade the distribution files to 4.6
- Upgrade custom extensions
 - 4.4 to 4.5
 - 4.5 to 4.6
- Upgrade the database to 4.6
 - 4.4 to 4.5
 - 4.5 to 4.6
- Regenerate the autoload array for extensions
- Run the system upgrade scripts for 4.6
 - 4.4 to 4.5
 - 4.5 to 4.6
- Additional upgrade step
- Enable ezjscore
- Enable admin2-design for back-end
- Clear the caches
- Upgrade extensions

Check for requirements

The eZ Components and PHP requirements

The minimum version required of eZ Components with eZ Publish 4.6 is "ezcomponents-ezp46", containing a fix to the package. This version is bundled in eZ Publish 4.6 Enterprise. eZ Publish 4.6 is compatible with PHP version 5.2.1 and above, but certified on RHEL 6 and Debian 6, PHP 5.3.x distros. See <http://ez.no/ezpublish/requirements> for more info.

Step 1: Upgrade the distribution files

The easiest way to upgrade the distribution files is to unpack eZ Publish 4.6 to a separate directory and then copy the directories that contain site-specific files from the existing installation. Make sure that you copy the following directories:

- design/example
- design/example_admin
- var
- settings/siteaccess
- settings/override
- (index_cluster.php)

Cluster note: If you are using cluster mode its necessary to copy index_cluster.php from the old version.

Replace "example" and "example_admin" with the actual names of your site accesses.

Important note: Because the new directory has replaced the original directory, the directory permissions need to be fixed. Use the following commands to do this.

(You have the choice between Shell commands or Alternative shell commands):

- **Shell commands**

These shell commands will give proper permission to the web server:

```
cd </path/to/your/eZ/Publish/directory>  
chmod -R a+rx design extension settings var
```

- **Alternative shell commands**

These commands will setup the permission more correctly, but require knowledge about the running web server (change 'nouser' to http server user).

```
cd </path/to/your/eZ/Publish/directory>  
chmod -R og+rx design extension settings var  
chown -R nouser:nouser design extension settings var
```

Step 2: Upgrade custom extensions

If you are using custom extensions, the sub-directories inside the "extension" directory will also have to be copied. However, make sure that you do not overwrite any extensions that are included in eZ Publish, which currently are eZ Flow (2.x), eZ Online Editor (5.x), eZ OpenOffice Document format (2.x), eZ JSCore (1.x), eZ Image Editor (1.x), eZ Comments (1.x), eZ Multiupload (1.x), eZ MB Password Expiry (1.x), eZ Network (1.x), eZ REST API Provider (1.x), eZ Script Monitor (1.x), eZ SI, eZ Find (2.x). Note that upgrading the distribution files will overwrite the autoload arrays for extensions. You will need to re-generate the autoload arrays for active extensions later.

See the dedicated upgrade instructions for eZ Flow and Website Interface below.

The updated versions of eZ Flow and Website Interface will also install the following extensions:

- eZ Website Toolbar
- eZ Star Rating
- eZ Google Maps Location

Note: For eZ Online Editor 5.x and eZ JS Core you will need to replace the following rewrite rules when using Virtual Hosts:

```
RewriteRule ^/var/cache/texttoimage/./* - [L]
RewriteRule ^/var/[~/]+/cache/(texttoimage|public)/.* - [L]
```

with:

```
RewriteRule ^/var/([~/]+)?cache/(texttoimage|public)/.* - [L]
```

For more detailed instructions, see the dedicated [eZ OE](#) and eZ JS Core doc pages.

Step 3: updateimagesystem.php

If the installation is a new installation or based on any eZ Publish version since 3.3 then you can skip running the "updateimagesystem.php" script. If not, then execute it using:

```
php update/common/scripts/4.1/updateimagesystem.php -s $SITE_ACCESS
```

Step 4: Upgrade the database

The update script for the database is located in the following locations, skip the ones that don't apply to the version your upgrading from:

```
<eZ Publish root>/update/database/<mysql|postgresql>/4.5/  
dbupdate-4.4.0-to-4.5.0.sql  
<eZ Publish root>/update/database/<mysql|postgresql>/4.6/  
dbupdate-4.5.0-to-4.6.0.sql
```

You can run these with the appropriate sql command line tool or application.

MySQL

The "ezmysql" database driver has been deprecated in 4.5 and it is recommended to use the "ezmysql" driver instead. This requires that the "mysqli" extension in PHP is enabled and can be archived by changing the driver in your override or siteaccesses "site.ini.append.php" settings to:

```
[DatabaseSettings]  
DatabaseImplementation=ezmysql
```

Cluster

The "Mysql DB/DFS Cluster" database back-end has not yet been deprecated, but it is recommended to use the same driver as the main database driver, so change your "file.ini.append.php" settings in override or siteaccesses to use "eZDFSFileHandlerMySQLiBackend" for DFS Cluster or "eZDBFileHandlerMysqliBackend" for DB Cluster.

Note: The "index_cluster.php" file should be updated to

```
define( 'STORAGE_BACKEND', 'dfsmysqli' )
```

If you use DB cluster, remove "dfs" from the string.

Step 5: Regenerate the autoload array for extensions

The autoload system also has some changes, for example the autoload array for extensions is now placed in var/autoload of your eZ Publish installation (along the class changes in extensions itself).

To regenerate the autoload array, execute the following script from the root of your eZ Publish directory:

```
php bin/php/ezpgenerateautoloads.php --extension
```

Step 6: Run the system upgrade scripts

4.4 to 4.5

For eZ Publish 4.4 which was upgraded from 4.3, run the following command from the root of your eZ Publish directory:

```
php update/common/scripts/4.5/updatesectionidentifier.php -s $SITE_ACCESS
```

Note: Skip this step if this your 4.4 installation is not an upgrade from 4.3. If you are uncertain, please run the script.

4.5 to 4.6

There are two system upgrade scripts that can be run for the eZ Publish 4.6 upgrade, *"remove-trashedimages.php"* and *"updateordernumber.php"*.

Script 1: If you want to ensure no trashed images remains on your setup because of image aliases not being restored when restoring object from trash (issue #017781), you do the following:

1. make sure that no items remains in trash for any users (images from objects in trash would be impossible to restore not doing so)
2. run the following cleanup script by executing this command:

```
$ php update/common/scripts/4.6/removetrashedimages.php
```

Note: It is safe to skip this test, this is only to free extra space.

Script 2: You need to run an upgrade script to update webshop order numbers issue (#018233) by running:

```
$ php update/common/scripts/4.6/updateordernumber.php
```

Note: If you dont have webshop or you dont have orders (table *"ezorder"*) in your eZ Publish, you dont have to run this script.

Step 7: Additional upgrade steps

Update *"override.ini.append.php"*

Content panes are not visible on the *"User accounts"* tab in the back-end. To fix it on an already-installed eZPublish, remove the following rules from your *"override.ini.append.php"* for your admin siteaccess:

```
[window_controls]
Source=window_controls.tpl
MatchFile=window_controls_user.tpl
Subdir=templates
Match[navigation_part_identifier]=ezusernavigationpart

[windows]
Source=windows.tpl
MatchFile=windows_user.tpl
Subdir=templates
Match[navigation_part_identifier]=ezusernavigationpart
```

... and clear the cache after removing. See separate step below.

Step 8: Enable ezjscore

Both the "admin2" design, "ezoe", "ezwebin", "ezflow" and several other extensions requires the ezjscore extension to be enabled, this is archived by editing your "settings/override/site.ini.append.php" settings file to make sure it looks like this:

```
<pre>
[ExtensionSettings]
ActiveExtensions []
ActiveExtensions []=ezjscore
</pre>
```

Step 9: Enable admin2 design for back-end

To enable admin2 open "site.ini" for your admin site access. For an eZ Flow site this would for instance be:

```
settings/siteaccess/ezflow_site_admin/site.ini.append.php
```

In this file include "AdditionalSiteDesignList[]=admin2" to your [DesignSettings]. It is important that you add the *admin2* design before the *admin* design, in order to let extensions that have admin templates to continue to work. On an eZ Flow site, it would mean changing the settings from:

```
[DesignSettings]
SiteDesign=ezflow_site_admin
AdditionalSiteDesignList []=admin
AdditionalSiteDesignList []=ezflow
```

to

```
[DesignSettings]
SiteDesign=ezflow_site_admin
AdditionalSiteDesignList []=admin2
AdditionalSiteDesignList []=admin
AdditionalSiteDesignList []=ezflow
```

When using a plain or eZ Webin install you should make the same changes. In this case you will of course have a different SiteDesign setting and no AdditionalSiteDesignList[]=ezflow line.

To enable users to change their Administration Interface preferences (Location and Re-Edit editing preferences), add the following line to the [Toolbar_admin_right] block in settings/siteaccess/<siteaccess_name>/toolbar.ini.append.php:

```
Tool []=admin_preferences
```

So the block will then look like like this:

```
[Toolbar_admin_right]
Tool []
Tool []=admin_current_user
Tool []=admin_bookmarks
Tool []=admin_preferences
```

Step 10: Clear the caches

Whenever an eZ Publish solution is upgraded, all caches must be cleared in a proper way. This should be done from within a system shell:

1. Navigate into the eZ Publish 4.5 directory.
2. Run the script using the following shell command:

```
php bin/php/ezcache.php --clear-all --purge
```

Purging ensures that the caches are physically removed. When the "--purge" parameter is not specified, the caches will be expired but not removed.

Sometimes the script is unable to clear all cache files because of restrictive file/directory permission settings. Make sure that all cache files have been cleared by inspecting the contents of the various cache sub-directories within the "var" directory (typically the "var/cache/" and "var/<name_of_siteaccess>/cache/" directories). If there are any cache files left, you need to remove them manually.

Step 11: Upgrade extensions

When using eZ Webin and eZ Flow, these extension will also need to be updated.

The first thing to do is to change the extensions permissions. An example on how this is done:

```
$chmod -R a+rx extension
```

4.4 to 4.5:

eZ Webin 1.7 to 1.8

or

eZ Flow 2.2 to 2.3

4.5 to 4.6:

eZ Webin 1.8 to 1.9

or

eZ Flow 2.3 to 2.4

1.4.3 Direct upgrading to 4.7 from 4.1, 4.2, 4.3, 4.4 and 4.5

Please note that there are various documentation options for the direct upgrading to eZ Publish 4.7. There are 4 separate upgrade kits containing the information for upgrading from versions 4.1, 4.2, 4.3, 4.4 and 4.5 to 4.7 of eZ Publish.

- Direct upgrading from 4.1 to 4.7 (page [79](#))
- Direct upgrading from 4.2 to 4.7 (page [90](#))
- Direct upgrading from 4.3 to 4.7 (page [100](#))
- Direct upgrading from 4.4 to 4.7 (page [109](#))
- Direct upgrading from 4.5 to 4.7 (page [118](#))

Direct upgrading from 4.1 to 4.7

This section describes how to upgrade your existing eZ Publish 4.1 installation directly to version 4.7. Make sure that you have a working backup of the site before you do the actual upgrade, and make sure the installation you are performing the upgrade on is offline.

Important upgrade notes:

The procedure for upgrading directly from version 4.1.x, 4.2, 4.3, 4.4 or 4.5 to 4.7 consists of the following steps:

- Upgrade the distribution files to 4.7
- Upgrade custom extensions
 - 4.1.0 to 4.2
 - 4.2 to 4.3
 - 4.3 to 4.4
 - 4.4 to 4.5
 - 4.5 to 4.6
 - 4.6 to 4.7
- Upgrade the database to 4.7
 - 4.1.0 to 4.2
 - 4.2 to 4.3
 - 4.3 to 4.4
 - 4.4 to 4.5
 - 4.5 to 4.6
 - 4.6 to 4.7
- Regenerate the autoload array for extensions
- Run the system upgrade scripts for 4.7
 - 4.1.0 to 4.2
 - 4.2 to 4.3
 - 4.3 to 4.4
 - 4.4 to 4.5
 - 4.5 to 4.6
 - 4.6 to 4.7
- Additional upgrade step

- Enable ezjscore
- Enable admin2-design for back-end
- Clear the caches
- Upgrade extensions

Check for requirements

The eZ Components and PHP requirements

The minimum version required of eZ Components with eZ Publish 4.7 is "ezcomponents-ezp47", containing a fix to the package. This version is bundled in eZ Publish 4.7 Enterprise. eZ Publish 4.7 is compatible with PHP version 5.3 and above, but certified on RHEL 6 and Debian 6, PHP 5.3.x distros. See eZ Publish Requirements for more info.

Step 1: Upgrade the distribution files

The easiest way to upgrade the distribution files is to unpack eZ Publish 4.7 to a separate directory and then copy the directories that contain site-specific files from the existing installation. Make sure that you copy the following directories:

- design/example
- design/example_admin
- var
- settings/siteaccess
- settings/override
- (index_cluster.php)

Cluster note: If you are using cluster mode it is necessary to copy index_cluster.php from the old version.

Replace "example" and "example_admin" with the actual names of your site accesses.

Important note: Because the new directory has replaced the original directory, the directory permissions need to be fixed. You have the choice between Shell commands or Alternative shell commands:

Use the following commands to do this:

- **Shell commands**

These shell commands will give proper permission to the web server:

```
cd </path/to/your/eZ/Publish/directory>
chmod -R a+rx design extension settings var
```

- **Alternative shell commands**

These commands will setup the permission more correctly, but require knowledge about the running web server (change 'nouser' to http server user).

```
cd </path/to/your/eZ/Publish/directory>
chmod -R og+rx design extension settings var
chown -R nouser:nouser design extension settings var
```

Step 2: Custom extensions

If you are using custom extensions, the sub-directories inside the "extension" directory will also have to be copied. However, make sure that you do not overwrite any extensions that are included in eZ Publish, which currently are eZ Flow (2.x), eZ Online Editor (5.x), eZ OpenOffice Document format (2.x), eZ JSCore (1.x), eZ Image Editor (1.x), eZ Comments (1.x), eZ Multiupload (1.x), eZ MB Password Expiry (1.x), eZ Network (1.x), eZ REST API Provider (1.x), eZ Script Monitor (1.x), eZ SI, eZ Find (2.x). Note that upgrading the distribution files will overwrite the autoload arrays for extensions. You will need to re-generate the autoload arrays for active extensions later.

See the dedicated upgrade instructions for eZ Flow and Website Interface below.

The updated versions of eZ Flow and Website Interface will also install the following extensions:

- eZ Website Toolbar
- eZ Star Rating
- eZ Google Maps Location

Note: For eZ Online Editor 5.x and eZ JS Core you will need to replace the following rewrite rules when using Virtual Hosts:

```
RewriteRule ^/var/cache/texttoimage/.* - [L]
RewriteRule ^/var/[^/]+/cache/(texttoimage|public)/.* - [L]
```

with:

```
RewriteRule ^/var/([^/]+)?cache/(texttoimage|public)/.* - [L]
```

For more detailed instructions, see the dedicated eZ OE and eZ JS Core doc pages.

Step 3: Upgrade the database

The update script for the database is located in the following locations. Skip the ones that don't apply to the version you are upgrading from:

```
<eZ Publish root>/update/database/<mysql|postgresql>/4.2/  
dbupdate-4.1.0-to-4.2.0.sql  
<eZ Publish root>/update/database/<mysql|postgresql>/4.3/  
dbupdate-4.2.0-to-4.3.0.sql  
<eZ Publish root>/update/database/<mysql|postgresql>/4.4/  
dbupdate-4.3.0-to-4.4.0.sql  
<eZ Publish root>/update/database/<mysql|postgresql>/4.5/  
dbupdate-4.4.0-to-4.5.0.sql  
<eZ Publish root>/update/database/<mysql|postgresql>/4.6/  
dbupdate-4.5.0-to-4.6.0.sql  
<eZ Publish root>/update/database/<mysql|postgresql>/4.6/  
dbupdate-4.6.0-to-4.7.0.sql
```

You can run these with the appropriate sql command line tool or application.

Note on 4.1.x to 4.2 upgrade: You need to selectively choose the correct sql from first file above if you are upgrading from 4.1.1, 4.1.2, 4.1.3 or 4.1.4.

If you are upgrading from 4.1.1: Ignore everything above -- END: from 4.1.1.

If you are upgrading from 4.1.2 or 4.1.3: Ignore everything above -- END: from 4.1.2.

If you are upgrading from 4.1.4: Ignore everything above -- END: from 4.1.4.

Clustering:

If you are upgrading from 4.1 or 4.2 using db or dfs cluster, make sure you apply the following sql updates (but for dfs make sure you change the table name from "ezdbfile" to "ezdfsfile"):

```
<eZ Publish root>/update/database/mysql/4.3/dbupdate-cluster-4.2.0-to-4.3.0.sql
```

Schema changes

Note: These schema changes are only necessary if you have eZ Flow installed and if you are making the following upgrades:

4.1.1 -> 4.7

4.1.2 -> 4.7

4.1.3 -> 4.7

4.1.4 -> 4.7

4.2 -> 4.7

You must run this sql in the mysql client:

```
ALTER TABLE ezm_pool ADD INDEX ezm_pool__block_id__ts_hidden ( block_id,  
ts_hidden );
```

MySQL

The "ezmysql" database driver has been deprecated in 4.5 and it is recommended to use the "ezmysqli" driver instead. This requires that the "mysqli" extension in PHP is enabled and can be archived by changing the driver in your override or siteaccesses "site.ini.append.php" settings to:

```
[DatabaseSettings]  
DatabaseImplementation=ezmysqli
```

Cluster

The "Mysql DB/DFS Cluster" database back-end has not yet been deprecated, but it is recommended to use the same driver as the main database driver, so change your "file.ini.append.php" settings in override or siteaccesses to use "eZDFSFileHandlerMySQLiBackend" for DFS Cluster or "eZDBFileHandlerMysqliBackend" for DB Cluster.

Note: The "index_cluster.php" file should be updated to

```
define( 'STORAGE_BACKEND', 'dfsmysqli' )
```

If you use DB cluster, remove "dfs" from the string.

Step 4: Regenerate the autoload array for extensions

The autoload system also has some changes, for example the autoload array for extensions is now placed in var/autoload of your eZ Publish installation (along the class changes in extensions itself).

To regenerate the autoload array, execute the following script from the root of your eZ Publish directory:

```
php bin/php/ezpgenerateautoloads.php --extension
```

Step 5: Run the system upgrade scripts

4.1 to 4.2

- correctxmlalign.php: after upgrading to 4.1 the default alignment for embedded images in the "ezxmltext" datatype is removed, this script will allow you to have the correct xml alignment.

```
php update/common/scripts/4.1/correctxmlalign.php -s $SITE_ACCESS
```

- `fixorphanimages.php`: this script will delete images from the storage directory that are no longer connected to any content object attribute

```
php update/common/scripts/4.2/fixorphanimages.php -s $SITE_ACCESS
```

4.2 to 4.3

This version includes a script to fix #15478: Node assignment is not removed when removing node from child list:

```
php update/common/scripts/4.3/updatesnodeassignment.php -s $SITE_ACCESS
```

4.3 to 4.4

Update script for this version contained a bug and is repeated in 4.4 to 4.5 below, thus it is not needed in this instruction.

4.4 to 4.5

For eZ Publish 4.4 which was upgraded from 4.3, run the following command from the root of your eZ Publish directory:

```
php update/common/scripts/4.5/updatesectionidentifier.php -s $SITE_ACCESS
```

Note: Skip this step if your 4.4 installation is not an upgrade from 4.3. If you are uncertain, please run the script.

4.5 to 4.6

There are two system upgrade scripts that can be run for the eZ Publish 4.6 upgrade, `remove-trashedimages.php` and `updateordernumber.php`.

Script 1: If you want to ensure no trashed images remains on your setup because of image aliases not being restored when restoring object from trash (issue #017781), you must do the following:

1. Make sure that no items remain in trash for any users (images from objects in trash would be impossible to restore not doing so)
2. Run the following cleanup script by executing this command:

```
$ php update/common/scripts/4.6/removetrashedimages.php
```

Note: It is safe to skip this test, this is only to free extra space.

Script 2: You need to run an upgrade script to update webshop order numbers issue (#018233) by running:

```
$ php update/common/scripts/4.6/updateordernumber.php
```

Note: If you don't have webshop or if you don't have orders (table "ezorder") in your eZ Publish, you don't have to run this script.

4.6 to 4.7

Mobile device detection

When upgrading from eZ Publish 4.6 to 4.7 and you are using eZ Flow, you can benefit from the new mobile device detection feature. Put the following settings in your "settings/override/site.ini.append.php" file:

```
[SiteAccessSettings]
[... ]
DetectMobileDevice=enabled
MobileSiteAccessURL=http://example.com/iphoneMobileSiteAccessList []
MobileSiteAccessList []=iphone
[... ]
```

An "iphone" siteaccess is provided out-of-the-box with your eZ Flow installation. Make sure that the MobileSiteAccessURL= points to the "iphone" site access. Note that this is a full URL and needs to include the http:// part.

The new object edit form in the Administration interface

While upgrading eZ Flow to the next release, you will have to replace the following override condition in "settings/siteaccess/<admin siteaccess>/override.ini.append.php". This is to adjust to the changes in the object edit form in the Administration interface.

```
[edit_frontpage]
Source=content/edit.tpl
MatchFile=edit/frontpage.tpl
Subdir=templates
Match[class_identifier]=frontpage
```

with this new one

```
[add_to_block_frontpage]
Source=content/parts/add_to_block.tpl
MatchFile=content/parts/add_to_block_frontpage.tpl
Subdir=templates
Match[class_identifier]=frontpage
```

The same operation may have to be done if the same override condition has been added for other class_identifier than "frontpage".

Step 6: Additional upgrade steps

Update "override.ini.append.php"

Content panes are not visible on the "User accounts" tab in the back-end. To fix it on an already-installed eZPublish, remove the following rules from your "override.ini.append.php" for your admin siteaccess:

```
[window_controls]
Source=window_controls.tpl
MatchFile=window_controls_user.tpl
Subdir=templates
Match[navigation_part_identifier]=ezusernavigationpart

[windows]
Source=windows.tpl
MatchFile=windows_user.tpl
Subdir=templates
Match[navigation_part_identifier]=ezusernavigationpart
```

... and clear the cache after removing. See separate step below.

Step 7: Enable ezjscore

Both the "admin2" design, "ezoe", "ezwebin", "ezflow" and several other extensions require the ezjscore extension to be enabled. This is archived by editing your "settings/override/site.ini.append.php" settings file to make sure it looks like this:

```
<pre>
[ExtensionSettings]
ActiveExtensions []
ActiveExtensions []=ezjscore
</pre>
```

Step 8: Enable admin2 design for back-end

To enable admin2 open "site.ini" for your admin site access. For an eZ Flow site this would be, for instance:

```
settings/siteaccess/ezflow_site_admin/site.ini.append.php
```

In this file include "AdditionalSiteDesignList[]=admin2" to your [DesignSettings]. It is important that you add the *admin2* design before the *admin* design, in order to let extensions that have admin templates to continue to work. On an eZ Flow site, it would mean changing the settings from:

```
[DesignSettings]
SiteDesign=ezflow_site_admin
AdditionalSiteDesignList[]=admin
AdditionalSiteDesignList[]=ezflow
```

to

```
[DesignSettings]
SiteDesign=ezflow_site_admin
AdditionalSiteDesignList[]=admin2
AdditionalSiteDesignList[]=admin
AdditionalSiteDesignList[]=ezflow
```

When using a plain or eZ Webin install you should make the same changes. In this case you will of course have a different SiteDesign setting and no AdditionalSiteDesignList[]=ezflow line.

To enable users to change their Administration Interface preferences (Location and Re-Edit editing preferences), add the following line to the [Toolbar_admin_right] block in settings/siteaccess/<siteaccess_name>/toolbar.ini.append.php:

```
Tool[]=admin_preferences
```

So the block will then look like like this:

```
[Toolbar_admin_right]
Tool[]
Tool[]=admin_current_user
Tool[]=admin_bookmarks
Tool[]=admin_preferences
```

Step 9: Clear the caches

Whenever an eZ Publish solution is upgraded, all caches must be cleared in a proper way. **Note:** You should not be a root user when you start the caching process (refer to <http://issues.ez.no/15823>). The caching should be done from within a system shell:

1. Navigate into the eZ Publish 4.5 directory.
2. Run the script using the following shell command:

```
php bin/php/ezcache.php --clear-all --purge
```

Purging ensures that the caches are physically removed. When the "--purge" parameter is not specified, the caches will be expired but not removed.

Sometimes the script is unable to clear all cache files because of restrictive file/directory permission settings. Make sure that all cache files have been cleared by inspecting the contents of the various cache sub-directories within the "var" directory (typically the "var/cache/" and "var/<name_of_siteaccess>/cache/" directories). If there are any cache files left, you need to remove them manually.

Step 10: Upgrade extensions

When using eZ Webin and eZ Flow, these extension will also need to be updated.

The first thing to do is to change the extensions permissions. An example on how this is done:

```
$chmod -R a+rx extension
```

4.1 to 4.2:

eZ Webin 1.4 to 1.5

or

eZ Flow 1.1 to 2.0

4.2 to 4.3:

eZ Webin 1.5 to 1.6

or

eZ Flow 2.0 to 2.1

4.3 to 4.4:

eZ Webin 1.6 to 1.7

or

eZ Flow 2.1 to 2.2

4.4 to 4.5:

eZ Webin 1.7 to 1.8

or
eZ Flow 2.2 to 2.3

4.5 to 4.6:

eZ Webin 1.8 to 1.9
or
eZ Flow 2.3 to 2.4

4.6 to 4.7:

eZ Webin 1.9 to 2.0
or
eZ Flow 2.4 to 2.5

5
1
Year
information

Direct upgrading from 4.2 to 4.7

This section describes how to upgrade your existing eZ Publish 4.2 installation directly to version 4.7. Make sure that you have a working backup of the site before you do the actual upgrade, and make sure the installation you are performing the upgrade on is offline.

Important upgrade notes:

The procedure for upgrading directly from version 4.2 to 4.7 consists of the following steps:

- Upgrade the distribution files to 4.7
- Upgrade custom extensions
 - 4.2 to 4.3
 - 4.3 to 4.4
 - 4.4 to 4.5
 - 4.5 to 4.6
 - 4.6 to 4.7
- Upgrade the database to 4.7
 - 4.2 to 4.3
 - 4.3 to 4.4
 - 4.4 to 4.5
 - 4.5 to 4.6
 - 4.6 to 4.7
- Regenerate the autoload array for extensions
- Run the system upgrade scripts for 4.7
 - 4.2 to 4.3
 - 4.3 to 4.4
 - 4.4 to 4.5
 - 4.5 to 4.6
 - 4.6 to 4.7
- Additional upgrade step
- Enable ezjscore
- Enable admin2-design for back-end
- Clear the caches
- Upgrade extensions

Check for requirements

The eZ Components and PHP requirements

The minimum version required of eZ Components with eZ Publish 4.7 is "ezcomponents-ezp47", containing a fix to the package. This version is bundled in eZ Publish 4.7 Enterprise. eZ Publish 4.7 is compatible with PHP version 5.3 and above, but certified on RHEL 6 and Debian 6, PHP 5.3.x distros. See eZ Publish Requirements for more info.

Step 1: Upgrade the distribution files

The easiest way to upgrade the distribution files is to unpack eZ Publish 4.7 to a separate directory and then copy the directories that contain site-specific files from the existing installation. Make sure that you copy the following directories:

- design/example
- design/example_admin
- var
- settings/siteaccess
- settings/override
- index_cluster.php if running cluster mode

Note: if your cluster index file is named this way, copy it under a new, temporary name. Migration of this file is explained later in the procedure.

Replace "example" and "example_admin" with the actual names of your site accesses.

Important note: Because the new directory has replaced the original directory, the directory permissions need to be fixed. *You have the choice between Shell commands or Alternative shell commands.*

Use the following commands to do this:

- **Shell commands**

These shell commands will give proper permission to the web server:

```
cd </path/to/your/eZ/Publish/directory>
chmod -R a+rwX design extension settings var
```

- **Alternative shell commands**

These commands will setup the permission more correctly, but require knowledge about the running web server (change 'nouser' to http server user).

```
cd </path/to/your/eZ/Publish/directory>
chmod -R og+rx design extension settings var
chown -R nouser:nouser design extension settings var
```

Step 2: Custom extensions

If you are using custom extensions, the sub-directories inside the "extension" directory will also have to be copied. However, make sure that you do not overwrite any extensions that are included in eZ Publish, which currently are eZ Flow (2.x), eZ Online Editor (5.x), eZ OpenOffice Document format (2.x), eZ JSCore (1.x), eZ Image Editor (1.x), eZ Comments (1.x), eZ Multiupload (1.x), eZ MB Password Expiry (1.x), eZ Network (1.x), eZ REST API Provider (1.x), eZ Script Monitor (1.x), eZ SI, eZ Find (2.x). Note that upgrading the distribution files will overwrite the autoload arrays for extensions. You will need to regenerate the autoload arrays for active extensions later.

See the dedicated upgrade instructions for eZ Flow and Website Interface below.

The updated versions of eZ Flow and Website Interface will also install the following extensions:

- eZ Website Toolbar
- eZ Star Rating
- eZ Google Maps Location

Note: For eZ Online Editor 5.x and eZ JS Core you will need to replace the following rewrite rules when using Virtual Hosts:

```
RewriteRule ^/var/cache/texttoimage/./* - [L]
RewriteRule ^/var/[^/]+/cache/(texttoimage|public)/.* - [L]
```

with:

```
RewriteRule ^/var/([~/]+)?cache/(texttoimage|public)/.* - [L]
```

For more detailed instructions, see the dedicated eZ OE and eZ JS Core doc pages.

Step 3: Upgrade the database

The update script for the database is located in the following locations. Skip the ones that don't apply to the version your upgrading from:

```
<eZ Publish root>/update/database/<mysql|postgresql>/4.3/  
dbupdate-4.2.0-to-4.3.0.sql  
<eZ Publish root>/update/database/<mysql|postgresql>/4.4/  
dbupdate-4.3.0-to-4.4.0.sql  
<eZ Publish root>/update/database/<mysql|postgresql>/4.5/  
dbupdate-4.4.0-to-4.5.0.sql  
<eZ Publish root>/update/database/<mysql|postgresql>/4.6/  
dbupdate-4.4.0-to-4.5.0.sql  
<eZ Publish root>/update/database/<mysql|postgresql>/4.6/  
dbupdate-4.6.0-to-4.7.0.sql
```

You can run these with the appropriate sql command line tool or application.

If you are using a DFS cluster, you must run the following query:

```
ALTER TABLE 'ezdfsfile' CHANGE 'datatype' 'datatype' VARCHAR(255);
```

and if you are using a DB cluster, you must run the following query:

```
ALTER TABLE 'ezdbfile' CHANGE 'datatype' 'datatype' VARCHAR(255);
```

Schema changes

Note: The schema changes are only necessary if you have eZ Flow installed and if you are making an upgrade from 4.2 -> 4.7. If so you must run this sql on the mysql client

```
ALTER TABLE ezm_pool ADD INDEX ezm_pool__block_id__ts_hidden ( block_id,  
ts_hidden );
```

MySQL

The "ezmysql" database driver has been deprecated in 4.5 and it is recommended to use the "ezmysqli" driver instead. This requires that the "mysqli" extension in PHP is enabled and can be archived by changing the driver in your override or siteaccesses "site.ini.append.php" settings to:

```
[DatabaseSettings]  
DatabaseImplementation=ezmysqli
```

Clustering

In version 4.7, the way the eZ Publish cluster is configured has completely changed. Until this version, a custom index_cluster.php file that contained the configuration had to be created, and

included the file actually delivering the assets. This method is deprecated. `index_cluster.php` now exists by default in the distribution, and cluster configuration is done in the custom `config.cluster.php` file (or `config.php`).

Upgrading will be as simple as replicating the configuration variables (up to a dozen) previously defined in `index_cluster.php`, and put them in `config.cluster.php`.

More details about these directives can be found in the DFS or DB clusters configuration instructions, and more in-depth information about the configuration directives can be found in the cluster configuration settings documentation.

Step 4: Regenerate the autoload array for extensions

The autoload system also has some changes, for example the autoload array for extensions is now placed in `var/autoload` of your eZ Publish installation (along with the class changes in extensions itself).

To regenerate the autoload array, execute the following script from the root of your eZ Publish directory:

```
php bin/php/ezpgenerateautoloads.php --extension
```

Step 5: Run the system upgrade scripts

4.2 to 4.3

This version includes a script to fix #15478: Node assignment is not removed when removing node from child list:

```
php update/common/scripts/4.3/updatenodeassignment.php -s $SITE_ACCESS
```

4.3 to 4.4

The update script for this version contained a bug and is repeated in 4.4 to 4.5 below, thus is not needed in this instruction.

4.4 to 4.5

For eZ Publish 4.4 which was upgraded from 4.3, run the following command from the root of your eZ Publish directory:

```
php update/common/scripts/4.5/updatesectionidentifier.php -s $SITE_ACCESS
```

Note: Skip this step if your 4.4 installation is not an upgrade from 4.3. If you are uncertain, please run the script.

4.5 to 4.6

There are two system upgrade scripts that can be run for the eZ Publish 4.6 upgrade, *"remove-trashedimages.php"* and *"updateordernumber.php"*.

Script 1: If you want to ensure no trashed images remain on your setup because of image aliases not being restored when restoring object from trash (issue #017781), you must do the following:

1. Make sure that no items remains in trash for any users (images from objects in trash would be impossible to restore not doing so)
2. Run the following cleanup script by executing this command:

```
$ php update/common/scripts/4.6/removetrashedimages.php
```

Note: It is safe to skip this test, this is only to free extra space.

Script 2: You need to run an upgrade script to update webshop order numbers issue (#018233) by running:

```
$ php update/common/scripts/4.6/updateordernumber.php
```

Note: If you dont have webshop or you dont have orders (table *"ezorder"*) in your eZ Publish, you dont have to run this script.

Step 6: Additional upgrade steps

Update *"override.ini.append.php"*

Content panes are not visible on the *"User accounts"* tab in the back-end. To fix it on an already-installed eZ Publish, remove the following rules from your *"override.ini.append.php"* for your admin siteaccess:

```
[window_controls]
Source=window_controls.tpl
MatchFile=window_controls_user.tpl
Subdir=templates
Match[navigation_part_identifier]=ezusernavigationpart

[windows]
Source=windows.tpl
MatchFile=windows_user.tpl
Subdir=templates
Match[navigation_part_identifier]=ezusernavigationpart
```

Deprecated function (ezi18n()) in eZ Starrating

There is a deprecated function (`ezi18n()`) in extension `eZ Starrating` for `eZ Publish` versions 4.2, 4.3, 4.4.

This will generate an error if you try to clear the cache, when doing a direct upgrade to 4.7.

You need to manually disable this extension:

```
edit settings/override/
site.ini.append.php and remove or comment the following line
ActiveExtensions[]=ezstarrating
```

You need to activate the extension again after upgrading the respective packages

... and clear the cache after disabling it. See separate step below.

Step 7: Enable admin2 design for back-end

To enable `admin2` open "site.ini" for your admin site access. For an `eZ Flow` site this would be, for instance:

```
settings/siteaccess/ezflow_site_admin/site.ini.append.php
```

In this file include "AdditionalSiteDesignList[]=admin2" to your [DesignSettings]. It is important that you add the `admin2` design before the `admin` design, in order to let extensions that have admin templates to continue to work. On an `eZ Flow` site, it would mean changing the settings from:

```
[DesignSettings]
SiteDesign=ezflow_site_admin
AdditionalSiteDesignList[]=admin
AdditionalSiteDesignList[]=ezflow
```

to

```
[DesignSettings]
SiteDesign=ezflow_site_admin
AdditionalSiteDesignList[]=admin2
AdditionalSiteDesignList[]=admin
AdditionalSiteDesignList[]=ezflow
```

When using a plain or `eZ Webin` install you should make the same changes. In this case you will of course have a different `SiteDesign` setting and no `AdditionalSiteDesignList[]=ezflow` line.

To enable users to change their Administration Interface preferences (Location and Re-Edit editing preferences), add the following line to the [Toolbar_admin_right] block in `settings/siteaccess/<siteaccess_name>/toolbar.ini.append.php`:

```
Tool []=admin_preferences
```

So the block will then look like this:

```
[Toolbar_admin_right]
Tool []
Tool []=admin_current_user
Tool []=admin_bookmarks
Tool []=admin_preferences
```

Step 8: Clear the caches

Whenever an eZ Publish solution is upgraded, all caches must be cleared in a proper way. **Note:** You should not be a root user when you start the caching process (refer to <http://issues.ez.no/15823>). The caching should be done from within a system shell:

1. Navigate into the eZ Publish 4.5 directory.
2. Run the script using the following shell command:

```
php bin/php/ezcache.php --clear-all --purge
```

Purging ensures that the caches are physically removed. When the "--purge" parameter is not specified, the caches will be expired but not removed.

Sometimes the script is unable to clear all cache files because of restrictive file/directory permission settings. Make sure that all cache files have been cleared by inspecting the contents of the various cache sub-directories within the "var" directory (typically the "var/cache/" and "var/<name_of_siteaccess>/cache/" directories). If there are any cache files left, you need to remove them manually.

Step 9: Upgrade extensions

The first thing to do is to change the extensions permissions. An example on how this is done:

```
$chmod -R a+rx extension
```

Mobile device detection

When upgrading from eZ Publish 4.6 to 4.7 and you are using eZ Flow, you can benefit from the new mobile device detection feature. Put the following settings in your "settings/override/site.ini.append.php" file:

```
[SiteAccessSettings]
[...]
DetectMobileDevice=enabled
MobileSiteAccessURL=http://example.com/iphone
MobileSiteAccessList []
MobileSiteAccessList []=iphone
[...]
```

An "iphone" siteaccess is provided out-of-the-box with your eZ Flow installation. Make sure that the MobileSiteAccessURL= points to the "iphone" site access. Note that this is a full URL and needs to include the http:// part.

New object edit form in the Administration interface

While upgrading eZ Flow to the next release, you will have to replace the following override condition in "settings/siteaccess/<admin siteaccess>/override.ini.append.php". This is to adjust to the changes in the object edit form in the Administration interface.

```
[edit_frontpage]
Source=content/edit.tpl
MatchFile=edit/frontpage.tpl
Subdir=templates
Match[class_identifier]=frontpage
```

with this new one

```
[add_to_block_frontpage]
Source=content/parts/add_to_block.tpl
MatchFile=content/parts/add_to_block_frontpage.tpl
Subdir=templates
Match[class_identifier]=frontpage
```

The same operation may have to be done if the same override condition has been added for other class_identifier than "frontpage".

eZ Flow & eZ Webin

When using eZ Webin and eZ Flow, these extensions will also need to be updated. Follow the dedicated upgrade instructions, depending on what version you are upgrading from.

4.2 to 4.3

eZ Webin 1.5 to 1.6

eZ Flow 2.0 to 2.1

4.3 to 4.4

eZ Webin 1.6 to 1.7
eZ Flow 2.1 to 2.2

4.4 to 4.5

eZ Webin 1.7 to 1.8
eZ Flow 2.2 to 2.3

4.5 to 4.6

eZ Webin 1.8 to 1.9
eZ Flow 2.3 to 2.4

4.6 to 4.7

eZ Webin 1.9 to 2.0
eZ Flow 2.4 to 2.5

Share Your Information

Direct upgrading from 4.3 to 4.7

This section describes how to upgrade your existing eZ Publish 4.3 installation directly to version 4.7. Make sure that you have a working backup of the site before you do the actual upgrade, and make sure the installation you are performing the upgrade on is offline.

Important upgrade notes:

The procedure for upgrading directly from version 4.3 to 4.7 consists of the following steps:

- Upgrade the distribution files to 4.7
- Upgrade custom extensions
 - 4.3.to 4.4
 - 4.4 to 4.5
 - 4.5 to 4.6
 - 4.6 to 4.7
- Upgrade the database to 4.7
 - 4.3 to 4.4
 - 4.4 to 4.5
 - 4.5 to 4.6
 - 4.6 to 4.7
- Regenerate the autoload array for extensions
- Run the system upgrade scripts for 4.7
 - 4.3 to 4.4
 - 4.4 to 4.5
 - 4.5 to 4.6
 - 4.6 to 4.7
- Additional upgrade step
- Enable ezjscore
- Enable admin2-design for back-end
- Clear the caches
- Upgrade extensions

Check for requirements

The eZ Components and PHP requirements

The minimum version required of eZ Components with eZ Publish 4.7 is "ezcomponents-ezp47", containing a fix to the package. This version is bundled in eZ Publish 4.7 Enterprise. eZ Publish 4.7 is compatible with PHP version 5.3 and above, but certified on RHEL 6 and Debian 6, PHP 5.3.x distros. See eZ Publish Requirements for more info.

Step 1: Upgrade the distribution files

The easiest way to upgrade the distribution files is to unpack eZ Publish 4.7 to a separate directory and then copy the directories that contain site-specific files from the existing installation. Make sure that you copy the following directories:

- design/example
- design/example_admin
- var
- settings/siteaccess
- settings/override
- index_cluster.php if running cluster mode

Note: if your cluster index file is named this way, copy it under a new, temporary name. Migration of this file is explained later in the procedure.

Replace "example" and "example_admin" with the actual names of your site accesses.

Important note: Because the new directory has replaced the original directory, the directory permissions need to be fixed. You have the choice between Shell commands or Alternative shell commands.

Use the following commands to do this:

- **Shell commands**

These shell commands will give proper permission to the web server:

```
cd </path/to/your/eZ/Publish/directory>
chmod -R a+rwX design extension settings var
```

- **Alternative shell commands**

These commands will setup the permission more correctly, but require knowledge about the running web server (change 'nouser' to http server user).

```
cd </path/to/your/eZ/Publish/directory>
chmod -R og+rwx design extension settings var
chown -R nouser:nouser design extension settings var
```

Step 2: Upgrade custom extensions

If you are using custom extensions, the sub-directories inside the "extension" directory will also have to be copied. However, make sure that you do not overwrite any extensions that are included in eZ Publish, which currently are eZ Flow (2.x), eZ Online Editor (5.x), eZ OpenOffice Document format (2.x), eZ JSCore (1.x), eZ Image Editor (1.x), eZ Comments (1.x), eZ Multiupload (1.x), eZ MB Password Expiry (1.x), eZ Network (1.x), eZ REST API Provider (1.x), eZ Script Monitor (1.x), eZ SI, eZ Find (2.x). Note that upgrading the distribution files will overwrite the autoload arrays for extensions. You will need to regenerate the autoload arrays for active extensions later.

See the dedicated upgrade instructions for eZ Flow and Website Interface below.

The updated versions of eZ Flow and Website Interface will also install the following extensions:

- eZ Website Toolbar
- eZ Star Rating
- eZ Google Maps Location

Note: For eZ Online Editor 5.x and eZ JS Core you will need to replace the following rewrite rules when using Virtual Hosts:

```
RewriteRule ^/var/cache/texttoimage/.* - [L]
RewriteRule ^/var/[^/]+/cache/(texttoimage|public)/.* - [L]
```

with:

```
RewriteRule ^/var/([~/]+)?cache/(texttoimage|public)/.* - [L]
```

For more detailed instructions, see the dedicated eZ OE and eZ JS Core doc pages.

Step 3: Upgrade the database

The update script for the database is located in the following locations. Skip the ones that don't apply to the version your upgrading from:

```
<eZ Publish root>/update/database/<mysql|postgresql>/4.4/  
dbupdate-4.3.0-to-4.4.0.sql  
<eZ Publish root>/update/database/<mysql|postgresql>/4.5/  
dbupdate-4.4.0-to-4.5.0.sql  
<eZ Publish root>/update/database/<mysql|postgresql>/4.6/  
dbupdate-4.5.0-to-4.6.0.sql  
<eZ Publish root>/update/database/<mysql|postgresql>/4.6/  
dbupdate-4.6.0-to-4.7.0.sql
```

You can run these with the appropriate sql command line tool or application.

If you are using a DFS cluster, you must run the following query:

```
ALTER TABLE 'ezdfsfile' CHANGE 'datatype' 'datatype' VARCHAR(255);
```

and if you are using a DB cluster, you must run the following query:

```
ALTER TABLE 'ezdbfile' CHANGE 'datatype' 'datatype' VARCHAR(255);
```

MySQL

The "ezmysql" database driver has been deprecated in 4.5 and it is recommended to use the "ezmysql" driver instead. This requires that the "mysql" extension in PHP is enabled and can be archived by changing the driver in your override or siteaccesses "site.ini.append.php" settings to:

```
[DatabaseSettings]  
DatabaseImplementation=ezmysql
```

Cluster

In version 4.7, the way the eZ Publish cluster is configured has completely changed. Until this version, a custom `index_cluster.php` file that contained the configuration had to be created, and included the file actually delivering the assets. This method is deprecated. `index_cluster.php` now exists by default in the distribution, and cluster configuration is done in the custom `config.cluster.php` file (or `config.php`).

Upgrading will be as simple as replicating the configuration variables (up to a dozen) previously defined in `index_cluster.php`, and put them in `config.cluster.php`.

More details about these directives can be found in the DFS or DB clusters configuration instructions, and more in-depth information about the configuration directives can be found in the cluster configuration settings documentation.

Step 4: Regenerate the autoload array for extensions

The autoload system also has some changes, for example the autoload array for extensions is now placed in var/autoload of your eZ Publish installation (along the class changes in extensions itself).

To regenerate the autoload array, execute the following script from the root of your eZ Publish directory:

```
php bin/php/ezpgenerateautoloads.php --extension
```

Step 5: Run the system upgrade scripts

4.3 to 4.4

The update script for this version contained a bug and is repeated in 4.4 to 4.5 below, thus it is not needed in this instruction.

4.4 to 4.5

For eZ Publish 4.4 which was upgraded from 4.3, run the following command from the root of your eZ Publish directory:

```
php update/common/scripts/4.5/updatesectionidentifier.php -s $SITE_ACCESS
```

Note: Skip this step if your 4.4 installation is not an upgrade from 4.3. If you are uncertain, please run the script.

4.5 to 4.6

There are two system upgrade scripts that can be run for the eZ Publish 4.6 upgrade, "*remove-trashedimages.php*" and "*updateordernumber.php*".

Script 1: If you want to ensure that no trashed images remain on your setup because of image aliases not being restored when restoring object from trash (issue #017781), you must do the following:

1. Make sure that no items remain in trash for any users (images from objects in trash would be impossible to restore not doing so)
2. Run the following cleanup script by executing this command:

```
$ php update/common/scripts/4.6/removetrashedimages.php
```

Note: It is safe to skip this test, this is only to free extra space.

Script 2: You need to run an upgrade script to update webshop order numbers issue (#018233) by running:

```
$ php update/common/scripts/4.6/updateordernumber.php
```

Note: If you don't have a webshop or if you don't have orders (table "ezorder") in your eZ Publish, you don't have to run this script.

4.6 to 4.7

Mobile device detection

When upgrading from eZ Publish 4.6 to 4.7 and you are using eZ Flow, you can benefit from the new mobile device detection feature. Put the following settings in your "settings/override/site.ini.append.php" file:

```
[SiteAccessSettings]
[...]
DetectMobileDevice=enabled
MobileSiteAccessURL=http://example.com/iphoneMobileSiteAccessList[]
MobileSiteAccessList[]=iphone
[...]
```

An "iphone" siteaccess is provided out-of-the-box with your eZ Flow installation. Make sure that the MobileSiteAccessURL= points to the "iphone" site access. Note that this is a full URL and needs to include the http:// part.

The new object edit form in the Administration interface

While upgrading eZ Flow to the next release, you will have to replace the following override condition in "settings/siteaccess/<admin siteaccess>/override.ini.append.php". This is to adjust to the changes in the object edit form in the Administration interface.

```
[edit_frontpage]
Source=content/edit.tpl
MatchFile=edit/frontpage.tpl
Subdir=templates
Match[class_identifier]=frontpage
```

with this new one

```
[add_to_block_frontpage]
Source=content/parts/add_to_block.tpl
MatchFile=content/parts/add_to_block_frontpage.tpl
Subdir=templates
Match[class_identifier]=frontpage
```

The same operation may have to be done if the same override condition has been added for other `class_identifier` than "frontpage".

Step 6: Additional upgrade steps

Update "override.ini.append.php"

Content panes are not visible on the "User accounts" tab in the back-end. To fix it on an already-installed eZ Publish, remove the following rules from your "override.ini.append.php" for your admin siteaccess:

```
[window_controls]
Source=window_controls.tpl
MatchFile=window_controls_user.tpl
Subdir=templates
Match[navigation_part_identifier]=ezusernavigationpart

[windows]
Source=windows.tpl
MatchFile=windows_user.tpl
Subdir=templates
Match[navigation_part_identifier]=ezusernavigationpart
```

Deprecated function (ezi18n()) in eZ Starrating

There is a deprecated function (ezi18n()) in extension eZ Starrating for eZ Publish versions 4.2, 4.3, 4.4.

This will generate an error if you try to clear the cache, when doing a direct upgrade to 4.7.

You need to manually disable this extension:

```
edit settings/override/
site.ini.append.php and remove or comment the following line
ActiveExtensions[]=ezstarrating
```

You need to activate the extension again after upgrading the respective packages

... and clear the cache after disabling it. See separate step below.

Step 7: Clear the caches

Whenever an eZ Publish solution is upgraded, all caches must be cleared in a proper way. This should be done from within a system shell:

1. Navigate into the eZ Publish 4.5 directory.
2. Run the script using the following shell command:

```
php bin/php/ezcache.php --clear-all --purge
```

Purging ensures that the caches are physically removed. When the "--purge" parameter is not specified, the caches will be expired but not removed.

Sometimes the script is unable to clear all cache files because of restrictive file/directory permission settings. Make sure that all cache files have been cleared by inspecting the contents of the various cache sub-directories within the "var" directory (typically the "var/cache/" and "var/<name_of_siteaccess>/cache/" directories). If there are any cache files left, you need to remove them manually.

Step 8: Upgrade extensions

When upgrading from eZ Publish 4.6 to 4.7 and you are using eZ Flow, you can benefit from the new mobile device detection feature. Put the following settings in your "settings/override/site.ini.append.php" file:

```
[SiteAccessSettings]
[...]
DetectMobileDevice=enabled
MobileSiteAccessURL=http://example.com/iphone
MobileSiteAccessList []
MobileSiteAccessList []=iphone
[...]
```

An "iphone" siteaccess is provided out-of-the-box with your eZ Flow installation. Make sure that the MobileSiteAccessURL= points to the "iphone" site access. Note that this is a full URL and needs to include the http:// part.

New object edit form in the Administration interface

While upgrading eZ Flow to the next release, you will have to replace the following override condition in "settings/siteaccess/<admin siteaccess>/override.ini.append.php". This is to adjust to the changes in the object edit form in the Administration interface.

```
[edit_frontpage]
Source=content/edit.tpl
MatchFile=edit/frontpage.tpl
Subdir=templates
Match[class_identifier]=frontpage
```

with this new one

```
[add_to_block_frontpage]
Source=content/parts/add_to_block.tpl
MatchFile=content/parts/add_to_block_frontpage.tpl
Subdir=templates
Match[class_identifier]=frontpage
```

The same operation may have to be done if the same override condition has been added for other `class_identifier` than "frontpage".

eZ Flow & eZ Webin

When using eZ Webin and eZ Flow, these extensions will also need to be updated. Follow the dedicated upgrade instructions, depending on what version you are upgrading from.

4.3 to 4.4

eZ Webin 1.6 to 1.7
eZ Flow 2.1 to 2.2

4.4 to 4.5

eZ Webin 1.7 to 1.8
eZ Flow 2.2 to 2.3

4.4 to 4.5

eZ Webin 1.8 to 1.9
eZ Flow 2.3 to 2.4

4.6 to 4.7

eZ Webin 1.9 to 2.0
eZ Flow 2.4 to 2.5

Direct upgrading from 4.4 to 4.7

This section describes how to upgrade your existing eZ Publish 4.4 installation directly to version 4.7. Make sure that you have a working backup of the site before you do the actual upgrade, and make sure the installation you are performing the upgrade on is offline.

Important upgrade notes:

The procedure for upgrading directly from version 4.4 to 4.7 consists of the following steps:

- Upgrade the distribution files to 4.7
- Upgrade custom extensions
 - 4.4 to 4.5
 - 4.5 to 4.6
 - 4.6 to 4.7
- Upgrade the database to 4.7
 - 4.4 to 4.5
 - 4.5 to 4.6
 - 4.6 to 4.7
- Regenerate the autoload array for extensions
- Run the system upgrade scripts for 4.7
 - 4.4 to 4.5
 - 4.5 to 4.6
 - 4.6 to 4.7
- Additional upgrade step
- Enable ezjscore
- Enable admin2-design for back-end
- Clear the caches
- Upgrade extensions

Check for requirements

The eZ Components and PHP requirements

The minimum version required of eZ Components with eZ Publish 4.7 is "ezcomponents-ezp47", containing a fix to the package. This version is bundled in eZ Publish 4.7 Enterprise. eZ Publish 4.7 is compatible with PHP version 5.3 and above, but certified on RHEL 6 and Debian 6, PHP 5.3.x distros. See eZ Publish Requirements for more info.

Step 1: Upgrade the distribution files

The easiest way to upgrade the distribution files is to unpack eZ Publish 4.7 to a separate directory and then copy the directories that contain site-specific files from the existing installation. Make sure that you copy the following directories:

- design/example
 - design/example_admin
 - var
 - settings/siteaccess
 - settings/override
 - index_cluster.php if running cluster mode
- Important: f your cluster index file is named this way, copy it under a new, temporary name. Migrating this file is explained later in the procedure.**

Replace "example" and "example_admin" with the actual names of your site accesses.

Important note: Because the new directory has replaced the original directory, the directory permissions need to be fixed. You have the choice between Shell commands or Alternative shell commands.

Use the following commands to do this:

- **Shell commands**

These shell commands will give proper permission to the web server:

```
cd </path/to/your/eZ/Publish/directory>
chmod -R a+rx design extension settings var
```

- **Alternative shell commands**

These commands will setup the permission more correctly, but require knowledge about the running web server (change 'nouser' to http server user).

```
cd </path/to/your/eZ/Publish/directory>
chmod -R og+rx design extension settings var
chown -R nouser:nouser design extension settings var
```

Step 2: Upgrade custom extensions

If you are using custom extensions, the sub-directories inside the "extension" directory will also have to be copied. However, make sure that you do not overwrite any extensions that are included in eZ Publish, which currently are eZ Flow (2.x), eZ Online Editor (5.x), eZ OpenOffice Document format (2.x), eZ JSCore (1.x), eZ Image Editor (1.x), eZ Comments (1.x), eZ Multiupload (1.x), eZ MB Password Expiry (1.x), eZ Network (1.x), eZ REST API Provider (1.x), eZ Script Monitor (1.x), eZ SI, eZ Find (2.x). Note that upgrading the distribution files will overwrite the autoload arrays for extensions. You will need to regenerate the autoload arrays for active extensions later.

See the dedicated upgrade instructions for eZ Flow and Website Interface below.

The updated versions of eZ Flow and Website Interface will also install the following extensions:

- eZ Website Toolbar
- eZ Star Rating
- eZ Google Maps Location

Note: For eZ Online Editor 5.x and eZ JS Core you will need to replace the following rewrite rules when using Virtual Hosts:

```
RewriteRule ^/var/cache/texttoimage/./* - [L]
RewriteRule ^/var/[^/]+/cache/(texttoimage|public)/.* - [L]
```

with:

```
RewriteRule ^/var/([^/]+)?cache/(texttoimage|public)/.* - [L]
```

For more detailed instructions, see the dedicated eZ OE and eZ JS Core doc pages.

Step 3: Upgrade the database

The update script for the database is located in the following locations. Skip the ones that don't apply to the version your upgrading from:

```
<eZ Publish root>/update/database/<mysql|postgresql>/4.5/
dbupdate-4.4.0-to-4.5.0.sql
<eZ Publish root>/update/database/<mysql|postgresql>/4.6/
dbupdate-4.5.0-to-4.6.0.sql
<eZ Publish root>/update/database/<mysql|postgresql>/4.6/
dbupdate-4.6.0-to-4.7.0.sql
```

You can run these with the appropriate sql command line tool or application.

If you are using a DFS cluster, you must run the following query:

```
ALTER TABLE 'ezdfsfile' CHANGE 'datatype' 'datatype' VARCHAR(255);
```

and if you are using a DB cluster, you must run the following query:

```
ALTER TABLE 'ezdbfile' CHANGE 'datatype' 'datatype' VARCHAR(255);
```

MySQL

The "ezmysql" database driver has been deprecated in 4.5 and it is recommended to use the "ezmysql" driver instead. This requires that the "mysqli" extension in PHP is enabled and can be archived by changing the driver in your override or siteaccesses "site.ini.append.php" settings to:

```
[DatabaseSettings]
DatabaseImplementation=ezmysql
```

Cluster

The "Mysql DB/DFS Cluster" database back-end has not yet been deprecated, but it is recommended to use the same driver as the main database driver, so change your "file.ini.append.php" settings in override or siteaccesses to use "eZDFSFileHandlerMySQLiBackend" for DFS Cluster or "eZDBFileHandlerMySQLiBackend" for DB Cluster.

Note: The "index_cluster.php" file should be updated to

```
define( 'STORAGE_BACKEND', 'dfsmysqli' )
```

If you use DB cluster, remove "dfs" from the string.

Cluster

In version 4.7, the way the eZ Publish cluster is configured has completely changed. Until this version, a custom index_cluster.php file had to be created, that contained the configuration, and included the file actually delivering the assets. This method is deprecated. index_cluster.php now exists by default in the distribution, and cluster configuration is done in the custom config.cluster.php file (or config.php).

Upgrading will be as simple as replicating the configuration variables (up to a dozen) previously defined in index_cluster.php, and put them in config.cluster.php.

More details about these directives can be found in the DFS or DB clusters configuration instructions, and more in-depth information about the configuration directives can be found in the cluster configuration settings documentation.

Step 4: Regenerate the autoload array for extensions

The autoload system also has some changes, for example the autoload array for extensions is now placed in `var/autoload` of your eZ Publish installation (along the class changes in extensions itself).

To regenerate the autoload array, execute the following script from the root of your eZ Publish directory:

```
php bin/php/ezpgenerateautoloads.php --extension
```

Step 5: Run the system upgrade scripts

4.4 to 4.5

For eZ Publish 4.4 which was upgraded from 4.3, run the following command from the root of your eZ Publish directory:

```
php update/common/scripts/4.5/updatesectionidentifier.php -s $SITE_ACCESS
```

Note: Skip this step if your 4.4 installation is not an upgrade from 4.3. If you are uncertain, please run the script.

4.5 to 4.6

There are two system upgrade scripts that can be run for the eZ Publish 4.6 upgrade, *"remove-trashedimages.php"* and *"updateordernumber.php"*.

Script 1: If you want to ensure that no trashed images remain on your setup because of image aliases not being restored when restoring object from trash (issue #017781), you must do the following:

1. Make sure that no items remain in trash for any users (images from objects in trash would be impossible to restore not doing so)
2. Run the following cleanup script by executing this command:

```
$ php update/common/scripts/4.6/removetrashedimages.php
```

Note: It is safe to skip this test, this is only to free extra space.

Script 2: You need to run an upgrade script to update webshop order numbers issue (#018233) by running:

```
$ php update/common/scripts/4.6/updateordernumber.php
```

Note: If you don't have webshop or you don't have orders (table "ezorder") in your eZ Publish, you don't have to run this script.

4.6 to 4.7

Mobile device detection

When upgrading from eZ Publish 4.6 to 4.7 and you are using eZ Flow, you can benefit from the new mobile device detection feature. Put the following settings in your "site.ini.append.php" file:

```
[SiteAccessSettings]
[...]
DetectMobileDevice=enabled
MobileSiteAccessURL=http://example.com/iphone
MobileSiteAccessList[]=iphone
[...]
```

An "iphone" siteaccess is provided out-of-the-box with your eZ Flow installation. Make sure that the MobileSiteAccessURL= points to the "iphone" site access. Note that this is a full URL and needs to include the http:// part.

The new object edit form in the Administration interface

While upgrading eZ Flow to the next release, you will have to replace the following override condition in "settings/siteaccess/<admin siteaccess>/override.ini.append.php". This is to adjust to the changes in the object edit form in the Administration interface.

```
[edit_frontpage]
Source=content/edit.tpl
MatchFile=edit/frontpage.tpl
Subdir=templates
Match[class_identifier]=frontpage
```

with this new one

```
[add_to_block_frontpage]
Source=content/parts/add_to_block.tpl
MatchFile=content/parts/add_to_block_frontpage.tpl
Subdir=templates
Match[class_identifier]=frontpage
```

The same operation may have to be done if the same override condition has been added for other `class_identifier` than "frontpage".

Step 6: Additional upgrade steps

Update "override.ini.append.php"

Content panes are not visible on the "User accounts" tab in the back-end. To fix it on an already-installed eZ Publish, remove the following rules from your "override.ini.append.php" for your admin siteaccess:

```
[window_controls]
Source=window_controls.tpl
MatchFile=window_controls_user.tpl
Subdir=templates
Match[navigation_part_identifier]=ezusernavigationpart

[windows]
Source=windows.tpl
MatchFile=windows_user.tpl
Subdir=templates
Match[navigation_part_identifier]=ezusernavigationpart
```

Deprecated function (ezi18n()) in eZ Starrating

There is a deprecated function (ezi18n()) in extension eZ Starrating for eZ Publish versions 4.2, 4.3, 4.4.

This will generate an error if you try to clear the cache, when doing a direct upgrade to 4.7.

You need to manually disable this extension:

```
edit settings/override/
site.ini.append.php and remove or comment the following line
ActiveExtensions[]=ezstarrating
```

You need to activate the extension again after upgrading the respective packages

... and clear the cache after disabling it. See separate step below.

Step 7: Clear the caches

Whenever an eZ Publish solution is upgraded, all caches must be cleared in a proper way. This should be done from within a system shell:

1. Navigate into the eZ Publish 4.5 directory.

2. Run the script using the following shell command:

```
php bin/php/ezcache.php --clear-all --purge
```

Purging ensures that the caches are physically removed. When the "--purge" parameter is not specified, the caches will be expired but not removed.

Sometimes the script is unable to clear all cache files because of restrictive file/directory permission settings. Make sure that all cache files have been cleared by inspecting the contents of the various cache sub-directories within the "var" directory (typically the "var/cache/" and "var/<name_of_siteaccess>/cache/" directories). If there are any cache files left, you need to remove them manually.

Step 8: Upgrade extensions

The first thing to do is to change the extensions permissions. An example on how this is done:

```
$chmod -R a+rx extension
```

Mobile device detection

When upgrading from eZ Publish 4.6 to 4.7 and you are using eZ Flow, you can benefit from the new mobile device detection feature. Put the following settings in your "site.ini.append.php" file:

```
[SiteAccessSettings]
[...]
DetectMobileDevice=enabled
MobileSiteAccessURL=http://m.example.com/
MobileSiteAccessList[]=ezresponsive
MobileSiteAccessList[]=ezdemo
[...]
```

An "iphone" siteaccess is provided out-of-the-box with your eZ Flow installation. Make sure that the MobileSiteAccessURL= points to the "iphone" site access. Note that this is a full URL and needs to include the http:// part.

New object edit form in the Administration interface

While upgrading eZ Flow to the next release, you will have to replace the following override condition in "settings/siteaccess/<admin siteaccess>/override.ini.append.php". This is to adjust to the changes in the object edit form in the Administration interface.

```
[edit_frontpage]
Source=content/edit.tpl
MatchFile=edit/frontpage.tpl
Subdir=templates
Match[class_identifier]=frontpage
```

with this new one

```
[add_to_block_frontpage]
Source=content/parts/add_to_block.tpl
MatchFile=content/parts/add_to_block_frontpage.tpl
Subdir=templates
Match[class_identifier]=frontpage
```

The same operation may have to be done if the same override condition has been added for other `class_identifier` than "frontpage".

eZ Flow & eZ Webin

When using eZ Webin and eZ Flow, these extensions will also need to be updated. Follow the dedicated upgrade instructions, depending on what version you are upgrading from.

4.4 to 4.5

eZ Webin 1.7 to 1.8
or
eZ Flow 2.2 to 2.3

4.5 to 4.6

eZ Webin 1.8 to 1.9
or
eZ Flow 2.3 to 2.4

4.6 to 4.7

eZ Webin 1.9 to 2.0
or
eZ Flow 2.4 to 2.5

Direct upgrading from 4.5 to 4.7

This section describes how to upgrade your existing eZ Publish 4.5 installation directly to version 4.7. Make sure that you have a working backup of the site before you do the actual upgrade, and make sure the installation you are performing the upgrade on is offline.

Important upgrade notes:

The procedure for upgrading directly from version 4.5 to 4.7 consists of the following steps:

- Upgrade the distribution files to 4.7
- Upgrade custom extensions
 - 4.5 to 4.6
 - 4.6 to 4.7
- Upgrade the database to 4.7
 - 4.5 to 4.6
 - 4.6 to 4.7
- Regenerate the autoload array for extensions
- Run the system upgrade scripts for 4.7
 - 4.5 to 4.6
 - 4.6 to 4.7
- Additional upgrade step
- Enable ezjscore
- Enable admin2-design for back-end
- Clear the caches
- Upgrade extensions

Check for requirements

The eZ Components and PHP requirements

The minimum version required of eZ Components with eZ Publish 4.7 is "ezcomponents-ezp47", containing a fix to the package. This version is bundled in eZ Publish 4.7 Enterprise. eZ Publish 4.7 is compatible with PHP version 5.3 and above, but certified on RHEL 6 and Debian 6, PHP 5.3.x distros. See <http://ez.no/ezpublish/requirements> for more info.

Step 1: Upgrade the distribution files

The easiest way to upgrade the distribution files is to unpack eZ Publish 4.7 to a separate directory and then copy the directories that contain site-specific files from the existing installation. Make sure that you copy the following directories:

- design/example
- design/example_admin
- var
- settings/siteaccess
- settings/override
- (index_cluster.php)

Cluster note: If you are using cluster mode its necessary to copy index_cluster.php from the old version.

Replace "example" and "example_admin" with the actual names of your site accesses.

Important note: Because the new directory has replaced the original directory, the directory permissions need to be fixed. You have the choice between Shell commands or Alternative shell commands.

Use the following commands to do this.

- **Shell commands**

These shell commands will give proper permission to the web server:

```
cd </path/to/your/eZ/Publish/directory>
chmod -R a+rx design extension settings var
```

- **Alternative shell commands**

These commands will setup the permission more correctly, but require knowledge about the running web server (change 'nouser' to http server user).

```
cd </path/to/your/eZ/Publish/directory>
chmod -R og+rx design extension settings var
chown -R nouser:nouser design extension settings var
```

Step 2: Upgrade custom extensions

If you are using custom extensions, the sub-directories inside the "extension" directory will also have to be copied. However, make sure that you do not overwrite any extensions that are included in eZ Publish, which currently are eZ Flow (2.x), eZ Online Editor (5.x), eZ OpenOffice Document format (2.x), eZ JSCore (1.x), eZ Image Editor (1.x), eZ Comments (1.x), eZ Multiupload (1.x), eZ MB Password Expiry (1.x), eZ Network (1.x), eZ REST API Provider (1.x), eZ Script Monitor (1.x), eZ SI, eZ Find (2.x). Note that upgrading the distribution files will overwrite the autoload arrays for extensions. You will need to regenerate the autoload arrays for active extensions later.

See the dedicated upgrade instructions for eZ Flow and Website Interface below.

The updated versions of eZ Flow and Website Interface will also install the following extensions:

- eZ Website Toolbar
- eZ Star Rating
- eZ Google Maps Location

Note: For eZ Online Editor 5.x and eZ JS Core you will need to replace the following rewrite rules when using Virtual Hosts:

```
RewriteRule ^/var/cache/texttoimage/./* - [L]
RewriteRule ^/var/[^/]+/cache/(texttoimage|public)/.* - [L]
```

with:

```
RewriteRule ^/var/([~/]+)?cache/(texttoimage|public)/.* - [L]
```

For more detailed instructions, see the dedicated eZ OE and eZ JS Core doc pages.

Step 3: Upgrade the database

The update script for the database is located in the following locations. Skip the ones that don't apply to the version you are upgrading from:

```
<eZ Publish root>/update/database/<mysql|postgresql>/4.6/
dbupdate-4.5.0-to-4.6.0.sql
<eZ Publish root>/update/database/<mysql|postgresql>/4.6/
dbupdate-4.6.0-to-4.7.0.sql
```

You can run these with the appropriate sql command line tool or application.

If you are using a DFS cluster, you must run the following query:

```
<span class="line">ALTER TABLE 'ezdfsfile' CHANGE 'datatype' 'datatype'  
VARCHAR(255);</  
span>
```

and if you are using a DB cluster, you must run the following query:

```
<span class="line">ALTER TABLE 'ezdbfile' CHANGE 'datatype' 'datatype'  
VARCHAR(255);</  
span>
```

MySQL

The "ezmysql" database driver has been deprecated in 4.5 and it is recommended to use the "ezmysql" driver instead. This requires that the "mysqli" extension in PHP is enabled and can be archived by changing the driver in your override or siteaccesses "site.ini.append.php" settings to:

```
[DatabaseSettings]  
DatabaseImplementation=ezmysql
```

Cluster

The "Mysql DB/DFS Cluster" database back-end has not yet been deprecated, but it is recommended to use the same driver as the main database driver, so change your "file.ini.append.php" settings in override or siteaccesses to use "eZDFSFileHandlerMySQLiBackend" for DFS Cluster or "eZDBFileHandlerMysqliBackend" for DB Cluster.

Note: The "index_cluster.php" file should be updated to:

```
define( 'STORAGE_BACKEND', 'dfsmysqli' )
```

If you use DB cluster, remove "dfs" from the string.

Step 4: Regenerate the autoload array for extensions

The autoload system also has some changes, for example the autoload array for extensions is now placed in var/autoload of your eZ Publish installation (along the class changes in extensions itself).

To regenerate the autoload array, execute the following script from the root of your eZ Publish directory:

```
php bin/php/ezpgenerateautoloads.php --extension
```

Step 5: Run the system upgrade scripts

4.5 to 4.6

There are two system upgrade scripts that can be run for the eZ Publish 4.6 upgrade, "remove-trashedimages.php" and "updateordernumber.php".

Script 1: If you want to ensure no trashed images remain on your setup because of image aliases not being restored when restoring object from trash (issue #017781), you do the following:

1. Make sure that no items remain in trash for any users (images from objects in trash would be impossible to restore not doing so)
2. Run the following cleanup script by executing this command:

```
$ php update/common/scripts/4.6/removetrashedimages.php
```

Note: It is safe to skip this test, this is only to free extra space.

Script 2: You need to run an upgrade script to update webshop order numbers issue (#018233) by running:

```
$ php update/common/scripts/4.6/updateordernumber.php
```

Note: If you dont have a webshop or if you dont have orders (table "ezorder") in your eZ Publish, you dont have to run this script.

4.6 to 4.7

Mobile device detection

When upgrading from eZ Publish 4.6 to 4.7 and you are using eZ Flow, you can benefit from the new mobile device detection feature. Put the following settings in your "settings/override/site.ini.append.php" file:

```
[SiteAccessSettings]
[...]
DetectMobileDevice=enabled
MobileSiteAccessURL=http://example.com/iphoneMobileSiteAccessList []
MobileSiteAccessList []=iphone
[...]
```

An "iphone" siteaccess is provided out-of-the-box with your eZ Flow installation. Make sure that the MobileSiteAccessURL= points to the "iphone" site access. Note that this is a full URL and needs to include the http:// part.

The new object edit form in the Administration interface

While upgrading eZ Flow to the next release, you will have to replace the following override condition in "settings/siteaccess/<admin siteaccess>/override.ini.append.php". This is to adjust to the changes in the object edit form in the Administration interface.

```
[edit_frontpage]
Source=content/edit.tpl
MatchFile=edit/frontpage.tpl
Subdir=templates
Match[class_identifier]=frontpage
```

with this new one

```
[add_to_block_frontpage]
Source=content/parts/add_to_block.tpl
MatchFile=content/parts/add_to_block_frontpage.tpl
Subdir=templates
Match[class_identifier]=frontpage
```

The same operation may have to be done if the same override condition has been added for other class_identifier than "frontpage".

Step 6: Additional upgrade steps

Update "override.ini.append.php"

Content panes are not visible on the "User accounts" tab in the back-end. To fix it on an already-installed eZ Publish, remove the following rules from your "override.ini.append.php" for your admin siteaccess:

```
[window_controls]
Source=window_controls.tpl
MatchFile=window_controls_user.tpl
Subdir=templates
Match[navigation_part_identifier]=ezusernavigationpart

[windows]
Source=windows.tpl
MatchFile=windows_user.tpl
Subdir=templates
Match[navigation_part_identifier]=ezusernavigationpart
```

... and clear the cache after removing. See separate step below.

Step 7: Clear the caches

Whenever an eZ Publish solution is upgraded, all caches must be cleared in a proper way. This should be done from within a system shell:

1. Navigate into the eZ Publish 4.5 directory.
2. Run the script using the following shell command:

```
php bin/php/ezcache.php --clear-all --purge
```

Purging ensures that the caches are physically removed. When the "--purge" parameter is not specified, the caches will be expired but not removed.

Sometimes the script is unable to clear all cache files because of restrictive file/directory permission settings. Make sure that all cache files have been cleared by inspecting the contents of the various cache sub-directories within the "var" directory (typically the "var/cache/" and "var/<name_of_siteaccess>/cache/" directories). If there are any cache files left, you need to remove them manually.

Step 8: Upgrade extensions

The first thing to do is to change the extensions permissions. An example on how this is done:

```
$chmod -R a+rx extension
```

Mobile device detection

When upgrading from eZ Publish 4.6 to 4.7 and you are using eZ Flow, you can benefit from the new mobile device detection feature. Put the following settings in your "settings/override/site.ini.append.php" file:

```
[SiteAccessSettings]
[...]
DetectMobileDevice=enabled
MobileSiteAccessURL=http://example.com/iphone
MobileSiteAccessList []
MobileSiteAccessList []=iphone
[...]
```

An "iphone" siteaccess is provided out-of-the-box with your eZ Flow installation. Make sure that the MobileSiteAccessURL= points to the "iphone" site access. Note that this is a full URL and needs to include the http:// part.

New object edit form in the Administration interface

While upgrading eZ Flow to the next release, you will have to replace the following override condition in "settings/siteaccess/<admin siteaccess>/override.ini.append.php". This is to adjust to the changes in the object edit form in the Administration interface.

```
[edit_frontpage]
Source=content/edit.tpl
MatchFile=edit/frontpage.tpl
Subdir=templates
Match[class_identifier]=frontpage
```

with this new one

```
[add_to_block_frontpage]
Source=content/parts/add_to_block.tpl
MatchFile=content/parts/add_to_block_frontpage.tpl
Subdir=templates
Match[class_identifier]=frontpage
```

The same operation may have to be done if the same override condition has been added for other class_identifier than "frontpage".

eZ Flow & eZ Webin

When using eZ Webin and eZ Flow, these extensions will also need to be updated. Follow the dedicated upgrade instructions, depending on what version you are upgrading from.

4.6 to 4.7:

eZ Webin 1.9 to 2.0
eZ Flow 2.4 to 2.5

1.4.4 Direct upgrading to 5.1 from 4.7

The various documentation options for the direct upgrading to eZ Publish 5.1:

- Direct upgrading from 4.7 to 5.1 (page [127](#))

Direct upgrading from 4.7 to 5.1

This section describes how to upgrade your existing eZ Publish 4.7 installation to version 5.1. Make sure that you have a working backup of the site before you do the actual upgrade, and make sure that the installation you are performing the upgrade on is offline.

Note on Paths:

- `<ezp5-root>`: The root directory where eZ Publish 5 is installed in, examples: `"/home/myuser/www/"` or `"/var/sites/ezpublish/"`
- `<ezp5-root>/ezpublish_legacy/`: Root directory of "Legacy" (aka "Legacy Stack", refers to the eZ Publish 4.x installation which is bundled with eZ Publish 5) normally inside `ezpublish_legacy/`, example: `"/home/myuser/www/ezpublish_legacy/"`

Important upgrade notes:

Before starting, please make sure that the DebugByUser legacy setting is disabled, as it will interfere with the login mechanism. This is a known issue, and you can get more details and the fix itself in the related Jira, EYP-21237.

If you still need to use the debug by user feature, make sure to apply the patch from EYP-21237 or request for an official patch from the customer support team, until the fix is not yet distributed in a service pack.

The procedure for upgrading directly from version 4.7 to 5.1 consists of the following steps:

1. Upgrade the distribution files
2. Upgrade custom extensions
3. Upgrade the database
4. Run scripts
5. Regenerate the autoload array for extensions
6. Generate eZ Publish 5 yml configuration & symlink assets5
7. Fix images outside the var dir
8. Clear the caches
9. Update rewrite rules
10. Upgrade extensions (site package)

Check for requirements

php 5.3.3 and higher is needed. Further information regarding system requirements can be found on Requirements Documentation Page.

Step 1: Upgrade the distribution files

The easiest way to upgrade the distribution files is to unpack eZ Publish 5.1 to a separate directory and then copy the directories that contain site-specific files from the existing 4.7 installation into `"<ezp5-root>/ezpublish_legacy/"`. Make sure you copy the following directories:

- design/example
- design/example_admin
- var
- settings/siteaccess
- settings/override

Replace "example" and "example_admin" with the actual names of your site accesses, however make sure that you do not overwrite any designs that are included in eZ Publish distribution: admin, base, standard, mysite & plain, and *avoid* coping admin2 design as it has been merged into admin as of 5.0.

Note: *Because the new directory has replaced the original directory, the directory permissions need to be fixed. You can choose between Shell commands or Alternative shell commands.*

Use *one* of the following two alternatives to do this:

- **Shell commands**

These shell commands will give proper permission to the web server and command line users:

```
cd /<ezp5-root>/
chmod -R a+rwX ezpublish/{cache,logs,config} \
    ezpublish_legacy/{design,extension,settings,var}
```

- **Recommended shell commands using setfacl**

These commands will setup the permission more correctly, but require use of setfacl which is not available on all systems (change 'www-data' to your web server user if your system uses something else):

```
cd /<ezp5-root>/
sudo setfacl -R -m u:www-data:rwX -m u:www-data:rwX \
    ezpublish/{cache,logs,config} \
    ezpublish_legacy/{design,extension,settings,var}
sudo setfacl -dR -m u:www-data:rwX -m u:'whoami':rwX \
    ezpublish/{cache,logs,config} \
    ezpublish_legacy/{design,extension,settings,var}
```

Cluster environments

For cluster environments the `config.php` from the legacy folder must be used as reference. In a clean installation you can find it at the path `ezpublish_legacy/config.php-RECOMMENDED`. Copy the file and configure it as desired. More details can be found in the Setting it up for an eZDFSFileHandler document, in the step 3.

Also, please be aware that the **eZDB file handler has been discontinued**, and doesn't make part of eZ Publish 5.1 or higher versions. If you were using eZDB in your eZ Publish 4.7 installation you will need to uninstall the eZDB cluster. You can use eZDFS instead, which you can install by following the Setting it up for an eZDFSFileHandler tutorial.

Note: Be aware that memcache needs to be used on cluster environments. Please refer to the documentation on confluence for more details about memcache and usage example.

If using FS2 file handler

If you were using eZFS2 in your eZ Publish 4.7 installation, please be aware that the **eZFS2 file handler has been discontinued**, and doesn't make part of eZ Publish 5.1 or higher versions. eZFS file handler will be used instead.

For that you will need to edit your `file.ini.append.php` and change the `FileHandler` setting, under the `[ClusteringSettings]` settings block, from `eZFS2FileHandler` to `eZFSFileHandler`. You can find an example below:

```
[ClusteringSettings]
FileHandler=eZFSFileHandler
```

Step 2: Upgrade custom extensions

If you are using custom extensions, the sub-directories inside the "extension" directory will also have to be copied from the existing 4.7 installation into `"/<ezp5-root>/ezpublish_legacy/extension/`". However, make sure that you do not overwrite any extensions that are included in eZ Publish distribution, which currently are (**Note:** *As of eZ Publish 5.0, these extensions have the same version number as eZ Publish*):

Extension name	Folder name	Note
eZ Flow	ezflow	
eZ JSCore	ezjscore	
eZ Online Editor	ezoe	
eZ OpenOffice Document	ezodf	

Format		
eZ Image Editor	ezie	
eZ Multiupload	ezmultiupload	
eZ MB Password Expiry	ezmbpaex	
eZ Network	ez_network	
eZ REST API Provider	ezprestapiprovider	
eZ Script Monitor	ezscriptmonitor	
eZ SI	ezsi	
eZ Find	ezfind	Optional install

Note that upgrading the distribution files will overwrite the autoload arrays for extensions. You will need to re-generate the autoload arrays for active extensions later.

Important: *If you plan to upgrade your eZ Website Interface, eZ Flow or eZ Demo site package as well, then additional extensions will be updated and the step for re-generate the autoload arrays can be skipped until that is done (links to documentation for upgrading these site packages can be found in the last step of this page).*

Step 3: Upgrade the database

The update scripts for the database is located in:

```
/<ezp5-root>/ezpublish_legacy/update/database/<mysql|postgresql>/5.0/
dbupdate-4.7.0-to-5.0.0.sql
/<ezp5-root>/ezpublish_legacy/update/database/<mysql|postgresql>/5.1/
dbupdate-5.0.0-to-5.1.0.sql
```

You can run these with the appropriate command line tool or application.

Step 4: Run scripts

The following script has to be run after the upgrade of the database:

```
cd /<ezp5-root>/ezpublish_legacy/
php update/common/scripts/5.0/deduplicatecontentstategrouplanguage.php
```

The following script must be executed to fix existing content and restore its relations.

Note: *This fix is about a known issue with relations and embedded/linked objects in XML text blocks that could be lost at object level for secondary languages and only applies to multilingual websites.*

```
php update/common/scripts/5.0/restorexmlrelations.php
```

The following script is responsible for disabling user accounts with suspicious user login (containing < and >). Remove the '--disable' option for a dry run:

```
php update/common/scripts/5.0/disableuspicioususers.php --disable
```

Note: If you are running `ezxmlexport`, an `ezxmlexport` upgrade SQL script have to be imported. The scripts for each supported database are available in `/<ezp5-root>/ezpublish_legacy/extension/ezxmlexport/update/database/[your database]/5.0/dbupdate-1.5-5.0.sql`.

Step 5: Regenerate the autoload array for extensions

To regenerate the autoload array, execute the following script from the root of your eZ Publish Legacy directory:

```
cd /<ezp5-root>/ezpublish_legacy/  
php bin/php/ezpgenerateautoloads.php --extension
```

Step 6: Generate eZ Publish 5 yml configuration & symlink assets

yml config

Note: This command will wipe out all custom yml config on subsequent runs, make sure to use `--backup` parameter if you run this command several times.

To generate yml configuration for the new Symfony stack, a console command has been provided to cover single site setups.

Perform the following command where `<group>` is the siteaccess group name (new concept, a grouping of siteaccesses that have something in common, db and admin siteaccess would be minimum determinators to be able to use the command), examples of group name: `'ezdemo_site'`, `'ezwebin_site'` or `'ezflow_site'`. And `<admin_siteaccess>` is the name of the admin siteaccess, for instance, `'ezdemo_site_admin'`.

```
cd /<ezp5-root>  
php ezpublish/console ezpublish:configure --env=prod <group> <admin-siteaccess>
```

assets

To be able to run eZ Publish 5 correctly, assets need to be exposed in the public `'web'` folder.

The following commands will first symlink eZ Publish 5 assets in `"Bundles"` and the second will symlink assets (design files like images, scripts and css, and files in var folder) from eZ Publish Legacy:

```
php ezpublish/console assets:install --symlink web  
php ezpublish/console ezpublish:legacy:assets_install --symlink web  
php ezpublish/console assetic:dump --env=prod web
```

Note: In both cases "web" is the default folder and can be skipped from the command. Further information about alternative options is available with `-h` just like it is with `"php ezpublish/console -h"`

Step 7: Fix images outside the var dir

This step is optional if you didn't changed the default location of eZ Publish's "var" dir. If you intend to use a different location for the "var" dir paths to images may need to be fixed. For that, please be sure to copy or move all the content from your initial var dir location, into your new var dir location.

Then, run the following script to fix the file locations:

```
cd /<ezp5-root>/ezpublish_legacy/  
php update/common/scripts/5.1/fiximagesoutsidevardir.php
```

IMPORTANT: This script only fixes the path to images. Binary files can be fixed on non clustered environments (FS) by moving the storage/original folder to the new location. This workaround does not apply to DB/DFS setups.

Step 8: Clear the caches

Whenever an eZ Publish solution is upgraded, all caches must be cleared in a proper way. This should be done from within a system shell:

1. Navigate into the new eZ Publish directory.
2. Run the script using the following shell command:

```
cd /<ezp5-root>/ezpublish_legacy/  
php bin/php/ezcache.php --clear-all --purge
```

Purging ensures that the caches are physically removed. When the "--purge" parameter is not specified, the caches will be expired but not removed.

Note: Sometimes the script is unable to clear all cache files because of restrictive file/directory permission settings. Make sure that all cache files have been cleared by inspecting the contents of the various cache sub-directories within the "var" directory (typically the `"var/cache/"` and `"var/<name_of_siteaccess>/cache/"` directories). If there are any cache files left, you need to remove them manually.

Step 9: Update rewrite rules

There are two ways eZ Publish 5 can be installed, either the full install with both the new Symfony stack and the legacy stack, or legacy only. In latter case you only need to point your '4.7 like'

rewrite rules to `/<ezp5-root>/ezpublish_legacy/` and that's it. Otherwise, update your virtual host according to the eZ Publish 5.1 rewrite rules on confluence and point your host configuration to `/<ezp5-root>/web/`.

Step 10: Upgrade Extensions (site package)

Next, depending on if you originally installed eZ Flow, eZ Webin or eZ Demo site, follow the steps mentioned in the eZ Webin, eZ Flow or eZ Demo upgrade documentation.

Known issue

For a list of known issues related to the 5.1 version, please be sure to check the related page on confluence.

1.5 Upgrading to 5.1

This chapter contains information related to the upgrade of an eZ Publish solution from version 5.0 to version 5.1.

Follow the upgrade procedure described in the following section:

- Upgrading from 5.0 to 5.1 (page [135](#))
 - Direct upgrading from 4.7 to 5.1 (page [127](#))

1.5.1 Upgrading from 5.0 to 5.1

This section describes how to upgrade your existing eZ Publish 5.0 installation to version 5.1. Make sure that you have a working backup of the site before you do the actual upgrade, and make sure that the installation you are performing the upgrade on is offline.

Note on Paths:

- `<ezp5-root>/`: The root directory where eZ Publish 5 is installed in, examples: `"/home/myuser/www/"` or `"/var/sites/ezpublish/"`
- `<ezp5-root>/ezpublish_legacy/`: Root directory of "Legacy" (aka "Legacy Stack", refers to the eZ Publish 4.x installation which is bundled with eZ Publish 5) normally inside `ezpublish_legacy/`, example: `"/home/myuser/www/ezpublish_legacy/"`

Important upgrade notes:

Before starting, please make sure that the DebugByUser legacy setting is disabled, as it will interfere with the login mechanism. This is a known issue, and you can get more details and the fix itself in the related Jira, EZP-21237.

If you still need to use the debug by user feature, make sure to apply the patch from EZP-21237 or request for an official patch from the customer support team, until the fix is not yet distributed in a service pack.

The procedure for upgrading directly from version 5.0 to 5.1 consists of the following steps:

1. Upgrade the distribution files
2. Upgrade custom extensions
3. Upgrade the database
4. Run scripts
5. Regenerate the autoload array for extensions
6. Generate eZ Publish 5 yml configuration & symlink assets5
7. Clear the caches
8. Upgrade extensions (site package)

Check for requirements

php 5.3.3 and higher is needed. Further information regarding system requirements can be found on Requirements Documentation Page.

Step 1: Upgrade the distribution files

The easiest way to upgrade the distribution files is to unpack eZ Publish 5.1 to a separate directory and then copy the directories that contain site-specific files from the existing 5.0 installation into `"<ezp5-root>/ezpublish_legacy/"`. Make sure you copy the following directories:

- design/example
- design/example_admin
- var
- settings/siteaccess
- settings/override

Replace "example" and "example_admin" with the actual names of your site accesses, however make sure that you do not overwrite any designs that are included in eZ Publish distribution: admin, base, standard, mysite & plain, and avoid copying admin2 design as it has been merged into admin as of 5.0.

Note: *Because the new directory has replaced the original directory, the directory permissions need to be fixed. You can choose between Shell commands or Alternative shell commands.*

Use *one* of the following two alternatives to do this:

- **Shell commands**

These shell commands will give proper permission to the web server and command line users:

```
cd /<ezp5-root>/
chmod -R a+rwX ezpublish/{cache,logs,config} \
    ezpublish_legacy/{design,extension,settings,var}
```

- **Recommended shell commands using setfacl**

These commands will setup the permission more correctly, but require use of setfacl which is not available on all systems (change 'www-data' to your web server user if your system uses something else):

```
cd /<ezp5-root>/
sudo setfacl -R -m u:www-data:rwX -m u:www-data:rwX \
    ezpublish/{cache,logs,config} \
    ezpublish_legacy/{design,extension,settings,var}
sudo setfacl -dR -m u:www-data:rwX -m u:'whoami':rwX \
    ezpublish/{cache,logs,config} \
    ezpublish_legacy/{design,extension,settings,var}
```

Cluster environments

For cluster environments the `config.php` from the legacy folder must be used as reference. In a clean installation you can find it at the path `ezpublish_legacy/config.php-RECOMMENDED`. Copy the file and configure it as desired. More details can be found in the Setting it up for an eZDFSFileHandler document, in the step 3.

Also, please be aware that the **eZDB file handler has been discontinued**, and doesn't make part of eZ Publish 5.1 or higher versions. If you were using eZDB in your eZ Publish 5.0 installation you will need to uninstall the eZDB cluster. You can use eZDFS instead, which you can install by following the Setting it up for an eZDFSFileHandler tutorial.

Note: Be aware that memcache needs to be used on cluster environments. Please refer to the documentation on confluence for more details about memcache and usage example.

If using FS2 file handler

If you were using eZFS2 in your eZ Publish 5.0 installation, please be aware that the **eZFS2 file handler has been discontinued**, and doesn't make part of eZ Publish 5.1 or higher versions. eZFS file handler will be used instead.

For that you will need to edit your `file.ini.append.php` and change the `FileHandler` setting, under the `[ClusteringSettings]` settings block, from `eZFS2FileHandler` to `eZFSFileHandler`. You can find an example below:

```
[ClusteringSettings]
FileHandler=eZFSFileHandler
```

Step 2: Upgrade custom extensions

If you are using custom extensions, the sub-directories inside the "extension" directory will also have to be copied from the existing 5.0 installation into `"/<ezp5-root>/ezpublish_legacy/extension/`". However, make sure that you do not overwrite any extensions that are included in eZ Publish distribution, which currently are (**Note:** *As of eZ Publish 5.1, these extensions have the same version number as eZ Publish*):

Extension name	Folder name	Note
eZ Flow	ezflow	
eZ JSCore	ezjscore	
eZ Online Editor	ezoe	
eZ OpenOffice Document	ezodf	

Format		
eZ Image Editor	ezie	
eZ Multiupload	ezmultiupload	
eZ MB Password Expiry	ezmbpaex	
eZ Network	ez_network	
eZ REST API Provider	ezprestapiprovider	
eZ Script Monitor	ezscriptmonitor	
eZ SI	ezsi	
eZ Find	ezfind	Optional install

Note that upgrading the distribution files will overwrite the autoload arrays for extensions. You will need to re-generate the autoload arrays for active extensions later.

Important: *If you plan to upgrade your eZ Website Interface, eZ Flow or eZ Demo site package as well, then additional extensions will be updated and the step for re-generate the autoload arrays can be skipped until that is done (links to documentation for upgrading these site packages can be found in the last step of this page).*

Step 3: Upgrade the database

The update script for the database is located in:

```
/<ezp5-root>/ezpublish_legacy/update/database/<mysql|postgresql>/5.1/dbupdate-5.0.0-to-5.1.0.sql
```

You can run this with the appropriate command line tool or application.

Step 4: Run scripts

No scripts needed to run for this upgrade.

Step 5: Regenerate the autoload array for extensions

To regenerate the autoload array, execute the following script from the root of your eZ Publish Legacy directory:

```
cd /<ezp5-root>/ezpublish_legacy/  
php bin/php/ezpgenerateautoloads.php --extension
```

Step 6: Generate eZ Publish 5 yml configuration & symlink assets

yml config

Warning: This command will wipe out all custom yml config you have done, make sure to use "--backup" parameter to make sure existing yml config is not deleted.

To generate yml configuration for the new Symfony stack, a console command has been provided to cover single site setups.

Perform the following command where '`<group>`' is the siteaccess group name (new concept, a grouping of siteaccesses that have something in common, db and admin siteaccess would be minimum determinators to be able to use the command), examples of group name: 'ezdemo_site', 'ezwebin_site' or 'ezflow_site'. And '`<admin_siteaccess>`' is the name of the admin siteaccess, for instance, 'ezdemo_site_admin'.

```
cd /<ezp5-root>
php eZpublish/console eZpublish:configure --env=prod <group> <admin-siteaccess>
```

assets

To be able to run eZ Publish 5 correctly, assets need to be exposed in the public 'web' folder.

The following commands will first symlink eZ Publish 5 assets in "Bundles" and the second will symlink assets (design files like images, scripts and css, and files in var folder) from eZ Publish Legacy:

```
php eZpublish/console assets:install --symlink web
php eZpublish/console eZpublish:legacy:assets_install --symlink web
php eZpublish/console assetic:dump --env=prod web
```

Note: In both cases "web" is the default folder and can be skipped from the command. Further information about alternative options is available with `-h` just like it is with `php eZpublish/console -h`

Step 7: Fix images outside the var dir

This step is optional if you didn't change the default location of eZ Publish's "var" dir.

If you intend to use a different location for the "var" dir paths to images may need to be fixed.

For that, please be sure to copy or move all the content from your initial var dir location, into your new var dir location.

Then, run the following script to fix the file locations:

```
cd /<ezp5-root>/eZpublish_legacy/
php update/common/scripts/5.1/fiximagesoutsidevardir.php
```

IMPORTANT: This script only fixes the path to images. Binary files can be fixed on non clustered environments (FS) by moving the storage/original folder to the new location. This workaround does not apply to DB/DFS setups.

Step 8: Clear the caches

Whenever an eZ Publish solution is upgraded, all caches must be cleared in a proper way. This should be done from within a system shell:

1. Navigate into the new eZ Publish directory.
2. Run the script using the following shell command:

```
cd /<ezp5-root>/ezpublish_legacy/  
php bin/php/ezcache.php --clear-all --purge
```

Purging ensures that the caches are physically removed. When the "--purge" parameter is not specified, the caches will be expired but not removed.

Note: Sometimes the script is unable to clear all cache files because of restrictive file/directory permission settings. Make sure that all cache files have been cleared by inspecting the contents of the various cache sub-directories within the "var" directory (typically the "var/cache/" and "var/<name_of_siteaccess>/cache/" directories). If there are any cache files left, you need to remove them manually.

Step 9: Update rewrite rules

There are two ways eZ Publish 5 can be installed, either the full install with both the new Symfony stack and the legacy stack, or legacy only. In latter case you only need to point your '4.7 like' rewrite rules to /<ezp5-root>/ezpublish_legacy/ and that's it. Otherwise, update your virtual host according to the eZ Publish 5.1 rewrite rules on confluence and point your host configuration to /<ezp5-root>/web/.

Step 10: Upgrade Extensions (site package)

Next, depending on if you originally installed eZ Flow, eZ Webin or eZ Demo site, follow the steps mentioned in the eZ Webin, eZ Flow or eZ Demo upgrade documentation.

Known issue

For a list of known issues related to the 5.1 version, please be sure to check the related page on confluence.

1.6 Upgrading to 5.0

This chapter contains information related to the upgrade of an eZ Publish solution from version 4.7 to version 5.0.

Follow the upgrade procedure described in the following section:

- [Upgrading from 4.7 to 5.0 \(page 142\)](#)

1.6.1 Upgrading from 4.7 to 5.0

This section describes how to upgrade your existing eZ Publish 4.7 installation to version 5.0. Make sure that you have a working backup of the site before you do the actual upgrade, and make sure that the installation you are performing the upgrade on is offline.

Note on Paths:

- `<ezp5-root>/`: The root directory where eZ Publish 5 is installed in, examples: `"/home/myuser/www/"` or `"/var/sites/ezpublish/"`
- `<ezp5-root>/ezpublish_legacy/`: Root directory of "Legacy" (aka "Legacy Stack", refers to the eZ Publish 4.x installation which is bundled with eZ Publish 5) normally inside `ezpublish_legacy/`, example: `"/home/myuser/www/ezpublish_legacy/"`

Important upgrade notes:

Before starting, please make sure that the `DebugByUser` legacy setting is disabled, as it will interfere with the login mechanism. This is a known issue, and you can get more details and the fix itself in the related Jira, [EZP-21237](#).

If you still need to use the debug by user feature, make sure to apply the patch from [EZP-21237](#) or request for an official patch from the customer support team, until the fix is not yet distributed in a service pack.

The procedure for upgrading directly from version 4.7 to 5.0 consists of the following steps:

1. Upgrade the distribution files
2. Upgrade custom extensions
3. Upgrade the database
4. Run scripts
5. Regenerate the autoload array for extensions
6. Generate eZ Publish 5 yml configuration & symlink assets
7. Clear the caches
8. Update rewrite rules
9. Upgrade extensions (site package)

Check for requirements

php 5.3.3 and higher is needed. Further information regarding system requirements can be found on [Requirements Documentation Page](#).

Step 1: Upgrade the distribution files

The easiest way to upgrade the distribution files is to unpack eZ Publish 5 to a separate directory and then copy the directories that contain site-specific files from the existing 4.7 installation into `"<ezp5-root>/ezpublish_legacy/"`. Make sure you copy the following directories:

- design/example
- design/example_admin
- var
- settings/siteaccess
- settings/override

Replace "example" and "example_admin" with the actual names of your site accesses, however make sure that you do not overwrite any designs that are included in eZ Publish distribution: admin, base, standard, mysite & plain, and avoid copying admin2 design as it has been merged into admin as of 5.0.

Note: *Because the new directory has replaced the original directory, the directory permissions need to be fixed. You can choose between Shell commands or Alternative shell commands.*

Use *one* of the following two alternatives to do this:

- **Shell commands**

These shell commands will give proper permission to the web server and command line users:

```
cd /<ezp5-root>/
chmod -R a+rwX ezpublish/{cache,logs,config} \
    ezpublish_legacy/{design,extension,settings,var}
```

- **Recommended shell commands using setfacl**

These commands will setup the permission more correctly, but require use of setfacl which is not available on all systems (change 'www-data' to your web server user if your system uses something else):

```
cd /<ezp5-root>/
sudo setfacl -R -m u:www-data:rwX -m u:www-data:rwX \
    ezpublish/{cache,logs,config} \
    ezpublish_legacy/{design,extension,settings,var}
sudo setfacl -dR -m u:www-data:rwX -m u:'whoami':rwX \
    ezpublish/{cache,logs,config} \
    ezpublish_legacy/{design,extension,settings,var}
```

Step 2: Upgrade custom extensions

If you are using custom extensions, the sub-directories inside the "extension" directory will also have to be copied from the existing 4.7 installation into `"/<ezp5-root>/ezpublish_legacy/extension/"`. However, make sure that you do not overwrite any extensions that are included in eZ Publish distribution, which currently are (**Note: As of eZ Publish 5.0, these extensions have the same version number as eZ Publish**):

Extension name	Folder name	Note
eZ Flow	ezflow	
eZ JSCore	ezjscore	
eZ Online Editor	ezoe	
eZ OpenOffice Document Format	ezodf	
eZ Image Editor	ezie	
eZ Multiupload	ezmultiupload	
eZ MB Password Expiry	ezmbpaex	
eZ Network	ez_network	
eZ REST API Provider	ezprestapiprovider	
eZ Script Monitor	ezscriptmonitor	
eZ SI	ezsi	
eZ Find	ezfind	Optional install

Note that upgrading the distribution files will overwrite the autoload arrays for extensions. You will need to re-generate the autoload arrays for active extensions later.

Important: *If you plan to upgrade your eZ Website Interface, eZ Flow or eZ Demo site package as well, then additional extensions will be updated and the step for re-generate the autoload arrays can be skipped until that is done (links to documentation for upgrading these site packages can be found in the last step of this page).*

Step 3: Upgrade the database

The update script for the database is located in:

```
/<ezp5-root>/ezpublish_legacy/update/database/<mysql|postgresql>/5.0/dbupdate-4.7.0-to-5.0.0.sql
```

You can run this with the appropriate command line tool or application.

Step 4: Run scripts

The following script has to be run after the upgrade of the database:

```
cd /<ezp5-root>/ezpublish_legacy/  
php update/common/scripts/5.0/deduplicatecontentstategrouplanguage.php
```

The following script must be executed to fix existing content and restore its relations.

Note: This fix is about a known issue with relations and embedded/linked objects in XML text blocks that could be lost at object level for secondary languages and only applies to multilingual websites.

```
php update/common/scripts/5.0/restorexmlrelations.php
```

The following script is responsible for disabling user accounts with suspicious user login (containing < and >). Remove the '--disable' option for a dry run:

```
php update/common/scripts/5.0/disableuspicioususers.php --disable
```

Note: If you are running `ezxmlexport`, an `ezxmlexport upgrade SQL` script have to be imported. The scripts for each supported database are available in `/<ezp5-root>/ezpublish_legacy/extension/ezxmlexport/update/database/[your database]/5.0/dbupdate-1.5-5.0.sql`.

Step 5: Regenerate the autoload array for extensions

The autoload system has undergone some changes also. For instance: the autoload array for extensions is now placed in `var/autoload` of your eZ Publish installation (along with the class changes in extensions itself).

To regenerate the autoload array, execute the following script from the root of your eZ Publish Legacy directory:

```
cd /<ezp5-root>/ezpublish_legacy/  
php bin/php/ezpgenerateautoloads.php --extension
```

Step 6: Generate eZ Publish 5 yml configuration & symlink assets

yml config

To generate yml configuration for the new Symfony stack, a console command has been provided to cover single site setups.

Perform the following command where '`<group>`' is the siteaccess group name (new concept, a grouping of siteaccesses that have something in common, db and admin siteaccess would be minimum determinators to be able to use the command), examples of group name: '`ezdemo_site`', '`ezwebin_site`' or '`ezflow_site`'. And '`<admin_siteaccess>`' is the name of the admin siteaccess, for instance, '`ezdemo_site_admin`'.

```
cd /<ezp5-root>
php ezpublish/console ezpublish:configure --env=prod <group> <admin-siteaccess>
```

assets

To be able to run eZ Publish 5 correctly, assets need to be exposed in the public 'web' folder.

The following commands will first symlink eZ Publish 5 assets in "Bundles" and the second will symlink assets (design files like images, scripts and css, and files in var folder) from eZ Publish Legacy:

```
php ezpublish/console assets:install --symlink web
php ezpublish/console ezpublish:legacy:assets_install --symlink web
```

Note: In both cases "web" is the default folder and can be skipped from the command. Further information about alternative options is available with `-h` just like it is with `php ezpublish/console -h`

Step 7: Clear the caches

Whenever an eZ Publish solution is upgraded, all caches must be cleared in a proper way. This should be done from within a system shell:

1. Navigate into the new eZ Publish directory.
2. Run the script using the following shell command:

```
cd /<ezp5-root>/ezpublish_legacy/
php bin/php/ezcache.php --clear-all --purge
```

Purging ensures that the caches are physically removed. When the "--purge" parameter is not specified, the caches will be expired but not removed.

Note: Sometimes the script is unable to clear all cache files because of restrictive file/directory permission settings. Make sure that all cache files have been cleared by inspecting the contents of the various cache sub-directories within the "var" directory (typically the "var/cache/" and "var/<name_of_siteaccess>/cache/" directories). If there are any cache files left, you need to remove them manually.

Step 8: Update rewrite rules

There are two ways eZ Publish 5 can be installed, either the full install with both the new Symfony stack and the legacy stack, or legacy only. In latter case you only need to point your '4.7 like' rewrite rules to `/<ezp5-root>/ezpublish_legacy/` and that's it. Otherwise, update your rewrite rules to resemble this and point it to `/<ezp5-root>/web/`.

Step 9: Upgrade Extensions (site package)

Next, depending on if you originally installed eZ Flow, eZ Webin or eZ Demo site, follow the steps mentioned in the eZ Webin, eZ Flow or eZ Demo upgrade documentation.

Known issue

Lazy sessions not supported

As Symfony always forces the creation of sessions, and regarding the way eZ Publish has this feature implemented, lazy sessions is currently not supported on eZ Publish 5.x.

Please note that the "ForceStart" setting still exists in eZ Publish 5 legacy, but it will not take any effect, since the creation of a session is always forced by Symfony.

Also please be aware that using this setting with Varnish or other similar caching systems may generate additional cache issues.

1.7 Upgrading to 4.7

This chapter contains information related to the upgrade of an eZ Publish solution to version 4.7.

Follow the upgrade procedure described in the following section:

- [Upgrading from 4.6 to 4.7 \(page 149\)](#)

1.7.1 Upgrading from 4.6 to 4.7

This section describes how to upgrade your existing eZ Publish 4.6 installation to version 4.7. Make sure that you have a working backup of the site before you do the actual upgrade, and make sure the installation you are performing the upgrade on is offline.

Important upgrade notes:

The procedure for upgrading directly from version 4.6 to 4.7 consists of the following steps:

1. Upgrade the distribution files to 4.7
2. Upgrade custom extensions
3. Upgrade the database to 4.7
4. Regenerate the autoload array for extensions
5. Clear the caches
6. Upgrade extensions

Check for requirements

The eZ Components and PHP requirements

The minimum version required of eZ Components with eZ Publish 4.7 is "ezcomponents-ezp47", containing a fix to the package. eZ Publish 4.7 is compatible with PHP version 5.2 and above. For more information regarding system requirements check out eZ Publish requirements.

Step 1: Upgrade the distribution files

The easiest way to upgrade the distribution files is to unpack eZ Publish 4.7 to a separate directory and then copy the directories that contain site-specific files from the existing installation. Make sure that you copy the following directories:

- design/example
- design/example_admin
- var
- settings/siteaccess
- settings/override

Replace "example" and "example_admin" with the actual names of your site accesses.

Important note: Because the new directory has replaced the original directory, the directory permissions need to be fixed. You have the choice between Shell commands or Alternative shell commands

Use the following commands to do this.

- **Shell commands**

These shell commands will give proper permission to the web server:

```
cd </path/to/your/eZ/Publish/directory>
chmod -R a+rwx design extension settings var
```

- **Alternative shell commands**

These commands will setup the permission more correctly, but require knowledge about the running web server.

```
chmod -R og+rwx design extension settings var
chown -R nouser:nouser design extension settings var
```

Step 2: Upgrade custom extensions

If you are using custom extensions, the sub-directories inside the "extension" directory will also have to be copied. However, make sure that you do not overwrite any extensions that are included in eZ Publish, which currently are eZ Flow (2.x) eZ Online Editor (5.x), eZ OpenOffice Document format (2.x), eZ JSCore (1.x), eZ Image Editor (1.x), eZ Comments (1.x), eZ Multiupload (1.x), eZ MB Password Expiry (1.x), eZ Network (1.x), eZ REST API Provider (1.x), eZ Script Monitor (1.x), eZ SI,), eZ Find (1.x). Note that upgrading the distribution files will overwrite the autoload arrays for extensions. You will need to re-generate the autoload arrays for active extensions later.

See the dedicated upgrade instructions for eZ Flow and Website Interface below.

The updated versions of eZ Flow and Website Interface will also install the following extensions:

- eZ Website Toolbar
- eZ Star Rating
- eZ Google Maps Location

Note: Both *admin2 design* and *webin (ezwebin extension)* require *ezjscore* extension to be enabled, look here for how to enable extensions.


```
-- START: from 4.6.0 using DFS cluster setup
ALTER TABLE 'ezdfsfile' CHANGE 'datatype' 'datatype' VARCHAR(255);
-- END: from 4.6.0 using DFS cluster setup
```

and if you are using a DB cluster, you must run the following query:

```
-- START: from 4.6.0 using DB cluster setup
ALTER TABLE 'ezdbfile' CHANGE 'datatype' 'datatype' VARCHAR(255);
-- END: from 4.6.0 using DB cluster setup
```

Note: The SQL script is located in eZ Publish distribution (update/database/mysql/4.7/).

Step 4: Regenerate the autoload array for extensions

The autoload system also has some changes, for example the autoload array for extensions is now placed in var/autoload of your eZ Publish installation (along the class changes in extensions itself).

To regenerate the autoload array, execute the following script from the root of your eZ Publish directory:

```
php bin/php/ezpgenerateautoloads.php --extension
```

Step 5: Clear the caches

Whenever an eZ Publish solution is upgraded, all caches must be cleared in a proper way. This should be done from within a system shell:

1. Navigate into the new eZ Publish directory.
2. Run the script using the following shell command:

```
php bin/php/ezcache.php --clear-all --purge
```

Purging ensures that the caches are physically removed. When the "--purge" parameter is not specified, the caches will be expired but not removed.

Sometimes the script is unable to clear all cache files because of restrictive file/directory permission settings. Make sure that all cache files have been cleared by inspecting the contents of the various cache sub-directories within the "var" directory (typically the "var/cache/" and "var/<name_of_siteaccess>/cache/" directories). If there are any cache files left, you need to remove them manually.

Step 6: Update rewrite rules

ezjscore extension previously had a file called `index_ajax.php` which was removed in 4.7. If you enabled this feature in the past you should now remove the following line or anything referring to `index_ajax.php` from your rewrite rules:

```
RewriteRule ^/([~/]+)?ezjscore/call.* /index_ajax\.php [L]
```

Step 7: Upgrade extensions

When using eZ Webin and eZ Flow, these extensions will also need to be updated.

The first thing to do is to change the extensions permissions. An example on how this is done:

```
$chmod -R a+rx extension
```

Mobile device detection

When upgrading from eZ Publish 4.6 to 4.7 and you are using eZ Flow, you can benefit from the new mobile device detection feature. Put the following settings in your `settings/override/site.ini.append.php` file:

```
[SiteAccessSettings]
[...]
DetectMobileDevice=enabled
MobileSiteAccessURL=http://example.com/iphone
MobileSiteAccessList []
MobileSiteAccessList []=iphone
[...]
```

An "iphone" siteaccess is provided out-of-the-box with your eZ Flow installation. Make sure that the `MobileSiteAccessURL=` points to the "iphone" site access. Note that this is a full URL and needs to include the `http://` part.

The new object edit form in the Administration interface

While upgrading eZ Flow to the next release, you will have to replace the following override condition in `settings/siteaccess/<admin siteaccess>/override.ini.append.php`. This is to adjust to the changes in the object edit form in the Administration interface.

```
[edit_frontpage]
Source=content/edit.tpl
MatchFile=edit/frontpage.tpl
Subdir=templates
```

```
Match[class_identifier]=frontpage
```

with this new one

```
[add_to_block_frontpage]
Source=content/parts/add_to_block.tpl
MatchFile=content/parts/add_to_block_frontpage.tpl
Subdir=templates
Match[class_identifier]=frontpage
```

The same operation may have to be done if the same override condition has been added for other class_identifier than "frontpage".

Next, depending on your own extensions, follow the steps mentioned in the eZ Webin and eZ Flow upgrade documentation.

1.8 Upgrading to 4.6

This chapter contains information related to the upgrade of an eZ Publish solution to version 4.6.

Follow the upgrade procedure described in the following section:

- [Upgrading from 4.5 to 4.6 \(page 156\)](#)

1.8.1 Upgrading from 4.5 to 4.6

This section describes how to upgrade your existing eZ Publish 4.5 installation to version 4.6. Make sure that you have a working backup of the site before you do the actual upgrade, and make sure the installation you are performing the upgrade on is offline.

Important upgrade notes:

The procedure for upgrading directly from version 4.5 to 4.6 consists of the following steps:

- Upgrade the distribution files to 4.6
- Upgrade custom extensions
- Upgrade the database to 4.6
- Regenerate the autoload array for extensions
- Run the system upgrade scripts for 4.6
- Additional upgrade step
- Clear the caches
- Upgrade extensions

Check for requirements

The eZ Components and PHP requirements

The minimum version required of eZ Components with eZ Publish 4.6 is "ezcomponents-ezp46", containing a fix to the package. eZ Publish 4.6 is compatible with PHP version 5.2 and above. For more information regarding system requirements check out eZ Publish requirements.

Step 1: Upgrade the distribution files

The easiest way to upgrade the distribution files is to unpack eZ Publish 4.6 to a separate directory and then copy the directories that contain site-specific files from the existing installation. Make sure that you copy the following directories:

- design/example
- design/example_admin
- var
- settings/siteaccess

- settings/override

Replace "example" and "example_admin" with the actual names of your site accesses.

Important note: Because the new directory has replaced the original directory, the directory permissions need to be fixed. Use the following commands to do this.

(You have the choice between Shell commands or Alternative shell commands):

- **Shell commands**

These shell commands will give proper permission to the web server:

```
cd </path/to/your/eZ/Publish/directory>
chmod -R a+rw design extension settings var
```

- **Alternative shell commands**

These commands will setup the permission more correctly, but require knowledge about the running web server.

```
chmod -R og+rw design extension settings var
chown -R nouser:nouser design extension settings var
```

Step 2: Upgrade custom extensions

If you are using custom extensions, the sub-directories inside the "extension" directory will also have to be copied. However, make sure that you do not overwrite any extensions that are included in eZ Publish, which currently are eZ Flow (2.x), eZ Online Editor (5.x), eZ OpenOffice Document format (2.x), eZ JSCore (1.x), eZ Image Editor (1.x), eZ Comments (1.x), eZ Multiupload (1.x), eZ MB Password Expiry (1.x), eZ Network (1.x), eZ REST API Provider (1.x), eZ Script Monitor (1.x), eZ SI,), eZ Find (1.x). Note that upgrading the distribution files will overwrite the autoload arrays for extensions. You will need to re-generate the autoload arrays for active extensions later.

See the dedicated upgrade instructions for eZ Flow and Website Interface below.

The updated versions of eZ Flow and Website Interface will also install the following extensions:

- eZ Website Toolbar
- eZ Star Rating
- eZ Google Maps Location

For eZ OE 5.x and eZ JS Core you will need to replace the following rewrite rules when using Virtual Hosts:

```
RewriteRule ^/var/cache/texttoimage/./* - [L]
RewriteRule ^/var/[^/]+/cache/(texttoimage|public)/.* - [L]
```

with:

```
RewriteRule ^/var/([^/]+)?cache/(texttoimage|public)/.* - [L]
```

Since `index_ajax.php` no longer exists in eZ JS Core you will need to remove the following rewrite rule when using Virtual Hosts:

```
RewriteRule ^/([^/]+)?ezjscore/call.* /index_ajax\.php [L]
```

or the rules in `.htaccess`:

```
RewriteRule ezjscore/call.* index_ajax.php
```

```
RewriteRule ^index_ajax\.php - [L]
```

For more detailed instructions, see the dedicated [eZ OE](#) and eZ JS Core doc pages.

Note: Both `admin2` design and `webin` (`ezwebin` extension) requires `ezjscore` extension to be enabled, look here for how to enabling extensions.

Step 3: Upgrade the database

The update script for the database is located in

```
<eZ Publish root>/update/database/<mysql|postgresql>/4.6/
dbupdate-4.5.0-to-4.6.0.sql
```

You can run this with the appropriate command line tool or application.

MySQL

The `"ezmysql"` database driver has been deprecated in 4.5 and it is recommended to use the `"ezmysql"` driver instead. This requires that the `"mysql"` extension in PHP is enabled and can be archived by changing the driver in your override or `siteaccesses "site.ini.append.php"` settings to:

```
[DatabaseSettings]
DatabaseImplementation=ezmysql
```

Cluster

The "Mysql DB/DFS Cluster" database back-end has not yet been deprecated, but it is recommended to use the same driver as the main database driver, so change your "file.ini.append.php" settings in override or siteaccesses to use "eZDFSFileHandlerMySQLiBackend" for DFS Cluster or "eZDBFileHandlerMySQLiBackend" for DB Cluster.

Note: The "index_cluster.php" file should be updated to

```
define( 'STORAGE_BACKEND', 'dfsmysqli' )
```

If you use DB cluster, remove "dfs" from the string.

Step 4: Regenerate the autoload array for extensions

The autoload system also has some changes, for example the autoload array for extensions is now placed in var/autoload of your eZ Publish installation (along the class changes in extensions itself).

To regenerate the autoload array, execute the following script from the root of your eZ Publish directory:

```
php bin/php/ezpgenerateautoloads.php --extension
```

Step 5: Run the system upgrade scripts

There are two system upgrade scripts that can be run for the eZ Publish 4.6 upgrade, "*remove-trashedimages.php*" and "*updateordernumber.php*".

Script 1: If you want to ensure no trashed images remains on your setup because of image aliases not being restored when restoring object from trash (issue #017781), you do the following:

1. make sure that no items remains in trash for any users (images from objects in trash would be impossible to restore not doing so)
2. run the following cleanup script by executing this command:

```
$ php update/common/scripts/4.6/removetrashedimages.php
```

Note: It is safe to skip this test, this is only to free extra space.

Script 2: You need to run an upgrade script to update webshop order numbers issue (#018233) by running:

```
$ php update/common/scripts/4.6/updateordernumber.php
```

Note: If you don't have webshop or you don't have orders (table "ezorder") in your eZ Publish, you don't have to run this script.

Step 6: Additional upgrade step

Update "override.ini.append.php"

Content panes are not visible on the "User accounts" tab in the back-end. To fix it on an already-installed eZ Publish, remove the following rules from your "override.ini.append.php" for your admin siteaccess:

```
[window_controls]
Source=window_controls.tpl
MatchFile=window_controls_user.tpl
Subdir=templates
Match[navigation_part_identifier]=ezusernavigationpart

[windows]
Source=windows.tpl
MatchFile=windows_user.tpl
Subdir=templates
Match[navigation_part_identifier]=ezusernavigationpart
```

... and clear the cache after removing. See separate step below.

Step 7: Clear the caches

Whenever an eZ Publish solution is upgraded, all caches must be cleared in a proper way. This should be done from within a system shell:

1. Navigate into the new eZ Publish directory.
2. Run the script using the following shell command:

```
php bin/php/ezcache.php --clear-all --purge
```

Purging ensures that the caches are physically removed. When the "--purge" parameter is not specified, the caches will be expired but not removed.

Sometimes the script is unable to clear all cache files because of restrictive file/directory permission settings. Make sure that all cache files have been cleared by inspecting the contents of the various cache sub-directories within the "var" directory (typically the "var/cache/" and "var/<name_of_siteaccess>/cache/" directories). If there are any cache files left, you need to remove them manually.

Step 8: Upgrade extensions

When using eZ Webin and eZ Flow, these extension will also need to be updated.

The first thing to do is to change the extensions permissions. An example on how this is done:

```
$chmod -R a+rwX extension
```

Next, depending on your own extensions, follow the steps mentioned in the eZ Webin and eZ Flow upgrade documentation.

1.9 Upgrading to 4.5

This chapter contains information related to the upgrade of an eZ Publish solution to version 4.5.

Follow the upgrade procedure described in the following section:

- [Upgrading from 4.4 to 4.5 \(page 163\)](#)

1.9.1 Upgrading from 4.4 to 4.5

This section describes how to upgrade your existing eZ Publish 4.4 installation to version 4.5. Make sure that you have a working backup of the site before you do the actual upgrade, and make sure the installation you are performing the upgrade on is offline.

Important upgrade notes:

The procedure for upgrading directly from version 4.4 to 4.5 consists of the following steps:

- Upgrade the distribution files to 4.5
- Upgrade custom extensions
- Upgrade the database to 4.5
- Regenerate the autoload array for extensions
- Run the system upgrade scripts for 4.5
- Clear the caches

Check for requirements

The eZ Components and PHP requirements

The minimum version required of eZ Components with eZ Publish 4.5 is "ezcomponents-ezp45", containing a fix to the package. eZ Publish 4.5 is compatible with PHP version 5.3 and above. For more information regarding system requirements check out <http://ez.no/ezpublish/requirements>

Step 1: Upgrade the distribution files

The easiest way to upgrade the distribution files is to unpack eZ Publish 4.5 to a separate directory and then copy the directories that contain site-specific files from the existing installation. Make sure that you copy the following directories:

- design/example
- design/example_admin
- var
- settings/siteaccess
- settings/override

Replace "example" and "example_admin" with the actual names of your site accesses.

Important note: Because the new directory has replaced the original directory, the directory permissions need to be fixed. Use the following commands to do this.

(You have the choice between Shell commands or Alternative shell commands):

- **Shell commands**

These shell commands will give proper permission to the web server:

```
cd </path/to/your/eZ/Publish/directory>
chmod -R a+rw design extension settings var
```

- **Alternative shell commands**

These commands will setup the permission more correctly, but require knowledge about the running web server.

```
chmod -R og+rw design extension settings var
chown -R nouser:nouser design extension settings var
```

Step 2: Upgrade custom extensions

If you are using custom extensions, the sub-directories inside the "extension" directory will also have to be copied. However, make sure that you do not overwrite any extensions that are included in eZ Publish, which currently are eZ Flow (2.x), eZ Online Editor (5.x), eZ OpenOffice Document format (2.x), eZ JSCore (1.x), eZ Image Editor (1.x), eZ Comments (1.x), eZ Multiupload (1.x), eZ MB Password Expiry (1.x), eZ Network (1.x), eZ REST API Provider (1.x), eZ Script Monitor (1.x), eZ SI,), eZ Find (1.x). Note that upgrading the distribution files will overwrite the autoload arrays for extensions. You will need to re-generate the autoload arrays for active extensions later.

See the dedicated upgrade instructions for eZ Flow and Website Interface below.

The updated versions of eZ Flow and Website Interface will also install the following extensions:

- eZ Website Toolbar
- eZ Star Rating
- eZ Google Maps Location

For eZ OE 5.x and eZ JS Core you will need to replace the following rewrite rules when using Virtual Hosts:

```
RewriteRule ^/var/cache/texttoimage/.+ - [L]
RewriteRule ^/var/[^/]+/cache/(texttoimage|public)/.+ - [L]
```

with:

```
RewriteRule ^/var/([^/]+)?cache/(texttoimage|public)/.+ - [L]
```

For more detailed instructions, see the dedicated [eZ OE](#) and [eZ JS Core](#) doc pages.

Note: Both admin2 design and webin (ezwebin extension) requires ezjscore extension to be enabled, look here for how to enabling extensions.

Step 3: Upgrade the database

The update script for the database is located in

```
<eZ Publish root>/update/database/<mysql|postgresql>/4.5/
dbupdate-4.4.0-to-4.5.0.sql
```

You can run this with the appropriate command line tool or application.

MySQL

The "ezmysql" database driver has been deprecated in 4.5 and it is recommended to use the "ezmysql" driver instead. This requires that the "mysql" extension in PHP is enabled and can be archived by changing the driver in your override or siteaccesses "site.ini.append.php" settings to:

```
[DatabaseSettings]
DatabaseImplementation=ezmysql
```

Cluster

The "Mysql DB/DFS Cluster" database back-end has not yet been deprecated, but it is recommended to use the same driver as the main database driver, so change your "file.ini.append.php" settings in override or siteaccesses to use "eZDFSFileHandlerMySQLBackend" for DFS Cluster or "eZDBFileHandlerMySQLBackend" for DB Cluster.

Note: The "index_cluster.php" file should be updated to

```
define( 'STORAGE_BACKEND', 'dfsmysql' )
```

If you use DB cluster, remove "dfs" from the string.

Step 4: Regenerate the autoload array for extensions

The autoload system also has some changes, for example the autoload array for extensions is now placed in var/autoload of your eZ Publish installation (along the class changes in extensions itself).

To regenerate the autoload array, execute the following script from the root of your eZ Publish directory:

```
php bin/php/ezpgenerateautoloads.php --extension
```

Step 5: Run the system upgrade scripts

For eZ Publish 4.4 which was upgraded from 4.3, run the following command from the root of your eZ Publish directory:

```
php update/common/scripts/4.5/updatesectionidentifier.php -s $SITE_ACCESS
```

Note: Skip this step if this your 4.4 installation is not an upgrade from 4.3. If you are uncertain, please run the script.

Step 6: Upgrade extensions

When using eZ Webin and eZ Flow, these extension will also need to be updated.

Note: If you have installed eZ Starrating, you have also to update this extension.

The first thing to do is to change the extensions permissions. An example on how this is done:

```
$chmod -R a+rx extension
```

Next, depending on your own extensions, follow the steps mentioned in the eZ Webin and eZ Flow upgrade documentation.

Step 7: Clear the caches

Whenever an eZ Publish solution is upgraded, all caches must be cleared in a proper way. This should be done from within a system shell:

1. Navigate into the eZ Publish 4.4 directory.
2. Run the script using the following shell command:

```
php bin/php/ezcache.php --clear-all --purge
```

Purging ensures that the caches are physically removed. When the "--purge" parameter is not specified, the caches will be expired but not removed.

Sometimes the script is unable to clear all cache files because of restrictive file/directory permission settings. Make sure that all cache files have been cleared by inspecting the contents of the various cache sub-directories within the "var" directory (typically the "var/cache/" and "var/<name_of_siteaccess>/cache/" directories). If there are any cache files left, you need to remove them manually.

1.10 Upgrading to 4.4

This chapter contains information related to the upgrade of an eZ Publish solution to version 4.4.

Follow the upgrade procedure described in the following section:

- [Upgrading from 4.3 to 4.4 \(page 169\)](#)

1.10.1 Upgrading from 4.3.x to 4.4.x

This section describes how to upgrade your existing eZ Publish 4.3.x installation to version 4.4. The upgrade procedure described below is generic and does not cover any specific cases (for example, running eZ Publish in a clustered environment).

Make sure that you have a working backup of the site before you do the actual upgrade.

Important upgrade notes:

The procedure for upgrading directly from version 4.3.x to 4.4 consists of the following steps:

- Upgrade the distribution files to 4.4
- Upgrade custom extensions
- Upgrade the database to 4.4
- Regenerate the autoload array for extensions
- Run the system upgrade scripts for 4.4
- Clear the caches

Check for requirements

The eZ Components 2009.1 requirements

With eZ Publish 4.4, eZ Components 2009.2.1 is the minimum version required. When downloading eZ Publish 4.4 it is possible to choose between a package bundled with eZ Components 2009.2.1, or a package without eZ Components. If you choose the latter, you must install eZ Components separately. For more information see <http://ezcomponents.org/docs/install>

The PHP requirements

eZ Publish 4.4 is compatible with PHP version 5.3. For more information regarding system requirements check out <http://ez.no/ezpublish/requirements>

Step 1: Upgrade the distribution files

The easiest way to upgrade the distribution files is to unpack eZ Publish 4.4 to a separate directory and then copy the directories that contain site-specific files from the existing installation. Make sure that you copy the following directories:

- design/example

- design/example_admin
- var
- settings/siteaccess
- settings/override

Replace "example" and "example_admin" with the actual names of your site accesses.

Important note

Because the new directory has replaced the original directory, the directory permissions need to be fixed. Use the following commands to do this.

(You have the choice between Shell commands or Alternative shell commands):

- **Shell commands**

These shell commands will give proper permission to the web server:

```
cd </path/to/your/eZ/Publish/directory>  
chmod -R a+rw design extension settings var
```

- **Alternative shell commands**

These commands will setup the permission more correctly, but require knowledge about the running web server.

```
chmod -R og+rw design extension settings var  
chown -R nouser:nouser design extension settings var
```

Step 2: Upgrade custom extensions

If you are using custom extensions, the sub-directories inside the "extension" directory will also have to be copied. However, make sure that you do not overwrite any extensions that are included in eZ Publish, which currently are eZ Online Editor (5.x), eZ OpenOffice.org (2.x) and eZ JSCore (1.x). Note that upgrading the distribution files will overwrite the autoload arrays for extensions. You will need to re-generate the autoload arrays for active extensions later.

See the dedicated upgrade instructions for eZ Flow and Website Interface below.

The updated versions of eZ Flow and Website Interface will also install the following extensions:

- eZ Website Toolbar

- eZ Star Rating
- eZ Google Maps Location

For eZ OE 5.x and eZ JS Core you will need to replace the following rewrite rules when using Virtual Hosts:

```
RewriteRule ^/var/cache/texttoimage/.*/.* - [L]
RewriteRule ^/var/[^/]+/cache/(texttoimage|public)/.* - [L]
```

with:

```
RewriteRule ^/var/([^/]+)/?cache/(texttoimage|public)/.* - [L]
```

For more detailed instructions, see the dedicated [eZ OE](#) and [eZ JS Core](#) doc pages.

Note: Both admin2 design and webin (ezwebin extension) requires ezjscore extension to be enabled, look here for how to enabling extensions.

Step 3: Upgrade the database

The update script for the database is located in

```
<eZ Publish root>/update/database/<mysql|postgresql>/4.4/
dbupdate-4.3.0-to-4.4.0.sql
```

You can run this with the appropriate command line tool or application.

Step 4: Regenerate the autoload array for extensions

The autoload system also has some changes, for example the autoload array for extensions is now placed in var/autoload of your eZ Publish installation (along the class changes in extensions itself).

To regenerate the autoload array, execute the following script from the root of your eZ Publish directory:

```
php bin/php/ezpgenerateautoloads.php --extension
```

Step 5: Run the system upgrade scripts

Run the following command from the root of your eZ Publish directory:

```
php update/common/scripts/4.4/updatesectionidentifier.php
```

Step 6: Clear the caches

Whenever an eZ Publish solution is upgraded, all caches must be cleared in a proper way. This should be done from within a system shell:

1. Navigate into the eZ Publish 4.4 directory.
2. Run the script using the following shell command:

```
php bin/php/ezcache.php --clear-all --purge
```

Purging ensures that the caches are physically removed. When the "--purge" parameter is not specified, the caches will be expired but not removed.

Sometimes the script is unable to clear all cache files because of restrictive file/directory permission settings. Make sure that all cache files have been cleared by inspecting the contents of the various cache sub-directories within the "var" directory (typically the "var/cache/" and "var/<name_of_siteaccess>/cache/" directories). If there are any cache files left, you need to remove them manually.

Step 7: Upgrading extensions

When using eZ Webin and eZ Flow, these extension will also need to be updated.

The first thing to do is to change the extensions permissions. An example on how this is done:

```
$chmod -R a+rw extension
```

Next, depending on your own extensions, follow the steps mentioned in the eZ Webin and eZ Flow upgrade documentation.

1.11 Upgrading to 4.3

This chapter contains information related to the upgrade of an eZ Publish solution to version 4.3. Follow the upgrade procedure described in the following section:

- [Upgrading from 4.2 to 4.3 \(page 174\)](#)

1.11.1 Upgrading from 4.2.x to 4.3.x

This section describes how to upgrade your existing eZ Publish 4.2.x installation to version 4.3. The upgrade procedure described below is generic and does not cover any specific cases (for example, running eZ Publish in a clustered environment).

Make sure that you have a working [backup](#) of the site before you do the actual upgrade.

Important upgrade notes:

The procedure for upgrading directly from version 4.2.x to 4.3 consists of the following steps:

1. Upgrade the distribution files to 4.3
2. Upgrade custom extensions
3. Upgrade the database to 4.3
4. Regenerate the autoload array for extensions
5. Run the system upgrade scripts for 4.3
6. (Optional) Enable Admin2 design for the eZ Publish Administration interface
7. Clear the caches

Check for requirements

The eZ Components 2009.1 requirements

With eZ Publish 4.2, **eZ Components 2009.2.1** is the minimum version required. When downloading eZ Publish 4.3 it is possible to choose between a package bundled with eZ Components 2009.2.1, or a package without eZ Components. If you choose the latter, you must install eZ Components separately. For more information see <http://ezcomponents.org/docs/install>

The PHP requirements

eZ Publish 4.3 is now also compatible with PHP version 5.3. For more information regarding system requirements check out <http://ez.no/ezpublish/requirements>

Step 1: Upgrade the distribution files

The easiest way to upgrade the distribution files is to unpack eZ Publish 4.3 to a separate directory and then copy the directories that contain site-specific files from the existing installation. Make sure that you copy the following directories:

- design/ *example*
- design/ *example_admin*
- var
- settings/siteaccess
- settings/override

Replace "example" and "example_admin" with the actual names of your site accesses.

Important note

Because the new directory has replaced the original directory, the directory permissions need to be fixed. Use the following commands to do this.

(You have the choice between Shell commands or Alternative shell commands):

- **Shell commands**

These shell commands will give proper permission to the web server:

```
cd </path/to/your/eZ/Publish/directory>
chmod -R a+rx design extension settings var
```

- **Alternative shell commands**

These commands will setup the permission more correctly, but require knowledge about the running web server.

```
chmod -R og+rx design extension settings var
chown -R nouser:nouser design extension settings var
```

Step 2: Custom extensions

If you are using custom extensions, the sub directories inside the "extension" directory will also have to be copied. However, make sure that you do not overwrite any extensions that are included in eZ Publish, which currently are eZ Online Editor (5.x), eZ OpenOffice.org (2.x) and eZ JSCore (1.x). Note that upgrading the distribution files will overwrite the autoload arrays for extensions. You will need to re-generate the autoload arrays for active extensions later.

See below for dedicated upgrade instructions for eZ Flow and eZ Webin.

The updated versions of eZ Flow and eZ Website Interface will also install the following extensions:

- eZ Website Toolbar
- eZ Star Rating
- eZ Google Maps Location

For eZ OE 5.x and eZ JS Core you will need to replace the following rewrite rules when using Virtual Hosts:

```
RewriteRule ^/var/cache/texttoimage/./* - [L]
RewriteRule ^/var/[^/]+/cache/(texttoimage|public)/.* - [L]
```

with:

```
RewriteRule ^/var/([^/]+)?cache/(texttoimage|public)/.* - [L]
```

For more detailed instructions, see the dedicated [eZ OE](#) and [eZ JS Core](#) doc pages.

Step 3: Upgrade the database

The update script for the database is located in

```
<eZP root>/update/database/<mysql|postgresql>/4.3/dbupdate-4.2.0-to-4.3.0.sql
```

You can run this with the appropriate command line tool or application

Step 4: Regenerate the autoload array for extensions

The autoload system also has some changes, for example the autoload array for extensions is now placed in var/autoload of your eZ Publish installation (along the class changes in extensions itself).

To regenerate the autoload array, execute the following script from the root of your eZ Publish directory:

```
php bin/php/ezpgenerateautoloads.php --extension
```

Step 5: Run the system upgrade scripts

Run the following command from the root of your eZ Publish directory:

```
php update/common/scripts/4.3/updatenodeassignment.php
```

Step 6: [optional] Enable admin2 design for eZ Publish Administration interface

To enable admin2 open site.ini for your admin site access. For an eZ Flow site this would for instance be:

```
settings/siteaccess/ezflow_site_admin/site.ini.append.php
```

In this file include `AdditionalSiteDesignList[]=admin2` to your `[DesignSettings]`. It is important that you add the admin2 design before the admin design, in order to let extensions that have admin templates to continue to work. On an eZ Flow site, it would mean changing the settings from:

```
[DesignSettings]
SiteDesign=ezflow_site_admin
AdditionalSiteDesignList []=admin
AdditionalSiteDesignList []=ezflow
```

to

```
[DesignSettings]SiteDesign=ezflow_site_admin
AdditionalSiteDesignList []=admin2
AdditionalSiteDesignList []=admin
AdditionalSiteDesignList []=ezflow
```

When using a plain or eZ Webin install you should make the same changes. In this case you will of course have a different `SiteDesign` setting and no `AdditionalSiteDesignList[]=ezflow` line.

To enable users to change their Administration Interface preferences (for example their preview tab preferences), add the following line to the `[Toolbar_admin_right]` block in `settings/siteaccess/<siteaccess_name>/toolbar.ini.append.php`:

```
Tool []=admin_preferences
```

So the block will then look like this:

```
[Toolbar_admin_right]
Tool []
Tool []=admin_current_user
Tool []=admin_bookmarks
Tool []=admin_preferences
```

Last step is to make sure `ezjscore` (extension) is enabled. For enabling extensions in general, look here, for 4.3 just make sure `ezjscore` is before `ezwebin` in `ActiveExtensions` if you use webin.

Step 7: Clear the caches

Whenever an eZ Publish solution is upgraded, all caches must be cleared in a proper way. **Note:** You should not be a root user when you start the caching process (refer to <http://issues.ez.no/15823>). The caching should be done from within a system shell:

1. Navigate into the eZ Publish 4.3 directory.
2. Run the script using the following shell command:

```
php bin/php/ezcache.php --clear-all --purge
```

Purging ensures that the caches are physically removed. When the "--purge" parameter is not specified, the caches will be expired but not removed.

Sometimes the script is unable to clear all cache files because of restrictive file/directory permission settings. Make sure that all cache files have been cleared by inspecting the contents of the various cache sub directories within the "var" directory (typically the "var/cache/" and "var/<name_of_siteaccess>/cache/" directories). If there are any cache files left, you need to remove them manually.

Upgrade eZ Flow and/or Website Interface separately

Please visit the eZ Flow and eZ Webin upgrade documentation for more info.

1.12 Upgrading to 4.2

This chapter contains information related to the upgrade of an eZ Publish solution to version 4.2. Follow the upgrade procedure that is described in the following section:

- [Upgrading from 4.1.x to 4.2.y](#)

1.12.1 Upgrading from 4.1.x to 4.2.y

This section describes how to upgrade your existing eZ Publish 4.1.x installation to version 4.2. If you are upgrading from version earlier than 4.0.3, you first need to upgrade to 4.0.3 before you can upgrade to 4.2 (Refer to either "[Upgrading from 3.10.x to 4.0.y](#)" or "[Upgrading from 4.0.x to 4.0.y](#)" depending on which version you are currently running.)

The upgrade procedure described below is generic and does not cover any specific cases (for example, running eZ Publish in a clustered environment).

Make sure that you have a working [backup](#) of the site before you do the actual upgrade.

Important upgrade notes:

You need to run "updateimagesystem.php" before the "dbupdate" scripts, otherwise the tables being upgraded will be removed.

The procedure for upgrading directly from version 4.1.x to 4.2 consists of the following steps:

1. Upgrading the distribution files to 4.2
2. Upgrading custom extensions
3. Updating image system
4. Upgrading the database to 4.2
5. Regenerate the autoload array for extensions
6. Running the system upgrade scripts for versions from 4.0.x to 4.2
7. Clearing the caches

Check for requirements

The eZ Components 2009.1 requirements

With eZ Publish 4.2, **eZ Components 2009.1** is the minimum version required. If you installed a previous version with PEAR, please upgrade to version 2009.1. For more information also see <http://ezcomponents.org/docs/install>

The PHP requirements

Although a PHP version 5.2.x is listed as required (with the exception of PHP 5.2.9 which has a serious regression in the array_unique() call), full PHP version 5.3 compatibility is also provided.

For more information regarding system requirements check out <http://ez.no/ezpublish/requirements>

Step 1: Upgrading the distribution files

The easiest way to upgrade the distribution files is to unpack eZ Publish 4.2 to a separate directory and then copy the directories that contain site-specific files from the existing installation. Make sure that you copy the following directories:

- design/ *example*
- design/ *example_admin*
- var
- settings/siteaccess
- settings/override

Replace "example" and "example_admin" with the actual names of your siteaccesses.

Step 2: Custom extensions

If you are using custom extensions, the subdirectories inside the "extension" directory will also have to be copied. However, make sure that you do not overwrite any extensions that come with eZ Publish, which currently are eZ Online Editor (5.0.4) and eZ OpenOffice.org (2.4.0). Note that upgrading the distribution files will overwrite the autoload arrays for extensions. You will need to re-generate the autoload arrays for active extensions later.

See below for dedicated upgrade instructions for eZ Flow and eZ Webin.

Note that ezdhtml is now replaced by ezoe, so you should not copy it over.

For eZ OE 5.x you will need to have the following rewrite rule if using Virtual Hosts:

```
RewriteRule ^/var/[~/]+/cache/public/.* - [L]
```

For more detailed instructions, see the dedicated doc page: http://ez.no/doc/extensions/online_editor/5_x/installation

The updated versions of eZ Flow and eZ Website Interface will also install the following extensions:

- eZ Website Toolbar (separated from eZ Web site)
- eZ JS Core (base library for client side Javascript and Ajax functions)
- eZ Star Rating
- eZ Google Maps Location

Step 3: updateimagesystem.php

This script is necessary to run if you have data of the 'ezimage' datatype, dating from before eZ Publish 3.3, in version 3.3 a new image system took over based on ezimagealiashandler.

If you are upgrading to the 4.2 series of eZ Publish for the first time, and the installation at hand has been running since prior to eZ Publish 3.3 then you need to run the updateimagesystem.php script before running any of the dbupdate scripts for version 4.2.

If the installation at hand is a new installation or based on any eZ Publish version since 3.3 then you can skip running the updateimagesystem.php script.

Step 4: Upgrading the database

The update script for the database is located in

```
<eZP root>/update/database/<mysql|postgresql>/4.2/dbupdate-4.1.0-to-4.2.0.sql
```

You can run this with the appropriate command line tool or application

Step 5: Regenerate the autoload array for extensions

The autoload system also has some changes, for example the autoload array for extensions is now placed in var/autoload of your eZ Publish installation (along the class changes in extensions itself).

To regenerate the autoload array, execute the following script from

```
php bin/php/ezpgenerateautoloads.php --extension
```

Step 6: Running the system upgrade scripts

Run the following two upgrade scripts:

The update script in <root of ez publish installation>/update/common/scripts/4.1 is:

- correctxmlalign.php: after upgrading to 4.1 the default alignment for embedded images in ezxmltext datatype is removed, this script will allow you to have the correct xmlalignment.

The update script in <root of ez publish installation>/update/common/scripts/4.2 is:

- fixorphanimages.php: this script will delete images from the storage directory that are no longer connected to any content object attribute

Step 7: Clearing the caches

Whenever an eZ Publish solution is upgraded, all caches must be cleared in a proper way. This should be done from within a system shell:

1. Navigate into the eZ Publish 4.2 directory.
2. Run the script using the following shell command:

```
php bin/php/ezcache.php --clear-all --purge
```

Purging ensures that the caches are physically removed. When the "--purge" parameter is not specified, the caches will be expired but not removed.

Sometimes the script is unable to clear all cache files because of restrictive file/directory permission settings. Make sure that all cache files have been cleared by inspecting the contents of the various cache sub-directories within the "var" directory (typically the "var/cache/" and "var/<name_of_siteaccess>/cache/" directories). If there are any cache files left, you need to remove them manually.

Upgrading eZ Flow and/or Website Interface separately

Step 1: Backup

Before you will start the Website Interface or eZ Flow upgrading process make sure that you have a backup of the existing website state including database, extensions, INI settings, etc. During the upgrade process existing eZ Flow and eZ Webin extensions will be removed and replaced with a new version. Website Interface and eZ Flow default content classes will be replaced as well. Upgrade script will also change some of the existing INI settings and add new ones which are required by latest version.

Step 2: eZ Flow upgrade

For information regarding upgrading to eZ Flow 2.0 please visit the following [extension upgrading page](#).

To get more information about available upgrade options, execute ezflowupgrade.php script with help parameter:

```
php bin/php/ezflowupgrade.php --help
```

Step 3: Re-write rules

Users running their eZ Flow sites in VH mode should update their extension related re-write rules to the following:

```
Rewriterule ^/extension/[^/]+/design/[^/]+/  
(stylesheets|flash|images|lib|javascripts?)/.* - [L]
```

Step 4: Website Interface upgrade

For more information regarding upgrading to eZ Webin 1.5 please visit the following [extension upgrading page](#).

To get more information about available upgrade options, execute ezwebinupgrade.php script with help parameter:

```
php bin/php/ezwebinupgrade.php --help
```

Step 5: Regenerate the autoload array for extensions

The autoload system also has some changes, for example the autoload array for extensions is now placed in var/autoload of your eZ Publish installation (along the class changes in extensions itself).

To regenerate the autoload array, execute the following script from:

```
php bin/php/ezpgenerateautoloads.php --extension
```

Step 6: Cache

Once the upgrade script is done, you need to clear the cache by executing following command from eZ Publish root folder:

```
php bin/php/ezcache.php --clear-all --purge
```

1.13 Upgrading to 4.1

This chapter contains information related to the upgrade of an eZ Publish solution to [version 4.1](#). Follow the upgrade procedures that are described in the following sections:

- [Upgrading from 4.0.x to 4.1.y \(page 190\)](#)
- [Upgrading from 4.1.x to 4.1.y \(page 186\)](#)

1.13.1 4.1.x to 4.1.y

This section describes how to upgrade your existing eZ Publish 4.1.x installation to version 4.1.y. If you are upgrading from a version prior to eZ Publish 4.1.0, refer to the "Upgrading from 4.0.x to 4.1.y (page 190)" section for upgrade instructions.

In order to benefit from the latest bug fixes, you should upgrade directly to the latest stable release in the 4.1 branch (4.1.2 at the time of writing). Refer to the [changelogs](#) and [security advisories](#) for more information about the issues that were fixed in the latest releases.

Important upgrade notes:

- "updateimagesystem.php" needs to be run before the "dbupdate" scripts, otherwise the tables being upgraded will be removed.

Please make a backup of your files and your database before proceeding. The procedure for upgrading directly from version 4.1.x to 4.1.y consists of the following steps:

1. Check eZ Components version
2. Update distribution files
3. Transfer custom extensions
4. Check updateimagesystem.php
5. Upgrade database
6. Regenerate autoload data for extensions
7. Run system upgrade scripts
8. Run eZ OE update script
9. Clear caches

Step 1: Check the eZ Components 2008.2.x requirements

With eZ Publish 4.1.1 and later, eZ Components 2008.2.2 is the minimum version required. If you installed a previous version with PEAR, please upgrade to version 2008.2.2. For more information about installing eZ Components see also <http://ezcomponents.org/docs/install>

Step 2: Updating the distribution files

The easiest way to upgrade the distribution files is to unpack eZ Publish 4.1.1 to a separate directory and then copy the directories that contain site-specific files from the existing installation. Make sure that you copy the following directories:

- design/example
- design/example_admin
- var
- settings/siteaccess
- settings/override

Replace ” *example*” and ” *example_admin*” with the actual names of your siteaccesses.

Step 3: Custom extensions

If you are using custom extensions, the sub directories inside the ”extension” directory will also have to be copied. However, make sure that you do not overwrite any extensions that come with eZ Publish (currently ”ezoe”, ”ezodf”). Note that upgrading the distribution files will overwrite the autoload arrays for extensions. You will need to re-generate the autoload arrays for active extensions later. See below for dedicated upgrade instructions for ez flow and ez webin.

Note that ezdhtml is now replaced by ezoe, so you should not copy it over.

For eZ OE 5.0.x, you will need to have the following rewrite rule if using Virtual Hosts:

```
RewriteRule ^/var/[^/]+/cache/public/.* - [L]
```

For more detailed instructions, see the dedicated doc page: http://ez.no/doc/extensions/online_editor/5_x/installation

Step 4: updateimagesystem.php

(If your upgrading from 4.1.0 and not directly from 4.0.x) First of all, if you don't have image data from before version 3.3 of eZ Publish, you don't have to do anything, you can skip running the ”updateimagesystem.php” script

If you do have old data or you suspect you might (installation is based on eZ Publish < 3.3), which you want to upgrade, then you have to import the tables ezimage and ezimagevariation from a database backup, as these tables were unfortunately removed in the dbupdate scripts of 4.1.0. After the tables have been restored, the updateimagesystem.php script needs to be run.

Step 5: Upgrading the database

The update script for the database is located in:

```
<eZP root>/update/database/<mysql|postgresql>/4.1/dbupdate-4.1.x-to-4.1.y.sql
```

You can run this with the appropriate command line tool or application

Step 6: Regenerate the autoload array for extensions

The autoload system has also some changes, for example the autoload array for extensions is now placed in `var/autoload` of your eZ Publish installation.

To regenerate the autoload array, execute the following script:

```
php bin/php/ezpgenerateautoloads.php --extension
```

Step 7: Running the system upgrade scripts

The update scripts are located in:

```
<eZP root>/update/common/scripts/4.1
```

Please note, that if these scripts were executed as part of the initial upgrade in 4.1.0, they do not have to be executed again this time.

- `addlockstategroup.php` (Used for creating locked states, part of the object states functionality. Updated in 4.1.1)
- `fixclassremoteid.php` (Fixing remote IDs of classes. Updated in 4.1.2 for PostgreSQL compatibility.)
- `fixobjectremoteid.php` (Fixes non-unique usage of content object remote IDs. Updated in 4.1.2 for PostgreSQL compatibility.)
- `correctxmlalign.php`: after upgrading to 4.1 the default alignment for embedded images in `ezxmltext` datatype was removed, this script will allow you to have the correct `xmlalign`ment.

Step 8: Run the eZ OE upgrade database script

(If you upgrade `eZOE` from 5.0.0 or earlier beta versions of the 5.0.x version) One of the bugs ([#14670](#)) fixed in OE 5.0.1 caused object relations to be stored with wrong relation type id, so they were not removed as relation when you removed them from editor area (`ezxmltext`). In order to fix the problem on existing relations created with OE 5.0.0 (also affected all prior beta- and rc-releases), apply the following:

```
<eZP root>/extension/eZOE/update/database/5.0/dbupdate-5.0.0-to-5.0.1.sql
```

Step 9: Clearing the caches

Whenever an eZ Publish solution is upgraded, all caches must be cleared in a proper way. This should be done from within a system shell:

Navigate into the eZ Publish 4.0.y directory.
Run the script using the following shell command:

```
php bin/php/ezcache.php --clear-all --purge
```

1.13.2 4.0.x to 4.1.y

This section describes how to upgrade your existing eZ Publish 4.0.x installation to version 4.1.y. If you are upgrading from version 4.0.3 or earlier, you need to first upgrade to 4.0.3 before you can upgrade to 4.1. (Refer to either "Upgrading from 3.10.x to 4.0.y (page 212)" or "Upgrading from 4.0.x to 4.0.y (page 200)" depending on which version you are currently running.)

The upgrade procedure described below is generic and does not cover any specific cases (for example, running eZ Publish in a clustered environment).

Make sure that you have a working backup of the site before you do the actual upgrade.

Important upgrade notes:

- `updateimagesystem.php` needs to be run before the `dbupdate` scripts, otherwise the tables being upgraded will be removed.

The procedure for upgrading directly from version 4.0.x to 4.1.y consists of the following steps:

1. Upgrading the distribution files to 4.1.y
2. Upgrading custom extensions
3. Updating image system
4. Upgrading the database to 4.1.y
5. Regenerate the autoload array for extensions
6. Running the system upgrade scripts for versions from 4.0.x to 4.1.y
7. Clearing the caches

Check the eZ Components 2008.2 requirements

With eZ Publish 4.1.0, **eZ Components 2008.2** is the minimum version required. If you installed a previous version with PEAR, please upgrade to version 2008.2. For more information see also <http://ezcomponents.org/docs/install>

Step 1: Upgrading the distribution files

The easiest way to upgrade the distribution files is to unpack eZ Publish 4.1.0 to a separate directory and then copy the directories that contain site-specific files from the existing installation. Make sure that you copy the following directories:

- `design/example`

- design/example_admin
- var
 - settings/siteaccess
 - settings/override

Replace "example" and "example_admin" with the actual names of your siteaccesses.

Step 2: Custom extensions

If you are using custom extensions, the subdirectories inside the "extension" directory will also have to be copied. However, make sure that you do not overwrite any extensions that come with eZ Publish (currently "ezoe", "ezodf"). Note that upgrading the distribution files will overwrite the autoload arrays for extensions. You will need to re-generate the autoload arrays for active extensions later.

See below for dedicated upgrade instructions for ez flow and ez webin.

Note that ezdhtml is now replaced by ezoe, so you should not copy it over.

For eZ OE 5.0, you will need to have the following rewrite rule if using Virtual Hosts:

```
RewriteRule ^/var/[^/]+/cache/public/.* - [L]
```

For extension development: when using modules always make sure that `$Result['path']` is assigned an array even if it's an empty one:

```
$Result['path'] = array();
```

For more detailed instructions, see the dedicated doc page: http://ez.no/doc/extensions/online_editor/5_x/installation

Step 3: updateimagesystem.php

This script is necessary to run if you have data of the 'ezimage' datatype, dating from before eZ Publish 3.3, in version 3.3 a new image system took over based on ezimagealias handler.

If you are upgrading to the 4.1.x series of eZ Publish for the first time, and the installation at hand have been running since eZ Publish < 3.3 then you need to run the updateimagesystem.php script before running any of the dbupdate scripts for version 4.1.0 or 4.1.1.

If the installation at hand is a new installation or based on any eZ Publish version >= 3.3 then you can skip running the updateimagesystem.php script.

Step 4: Upgrading the database

The update script for the database is located in

```
<eZP root>/update/database/<mysql|postgresql>/4.1/dbupdate-4.0.0-to-4.1.0.sql
```

You can run this with the appropriate command line tool or application

Step 5: Regenerate the autoload array for extensions

The autoload system has also some changes, for example the autoload array for extensions is now placed in var/autoload of your eZ Publish installation (along with the class changes in extensions itself).

To regenerate the autoload array, execute the following script from

```
php bin/php/ezpgenerateautoloads.php --extension
```

Step 6: Running the system upgrade scripts

The update scripts are located in <root of ez publish installation>/update/common/scripts/4.1

- `addlockstategroup.php`: is used for creating locked states, part of the object states functionality. It might be useful to know that you can only execute this script once. If you wish to run it a second time, it must be done like this:

```
truncate ezcobj_state ;
truncate ezcobj_state_group ;
truncate ezcobj_state_group_language ;
truncate ezcobj_state_language ;
truncate ezcobj_state_link ;

php update/common/scripts/4.1/addlockstategroup.php -sezwebin_site_admin
```

- `fixclassremoteid.php`: is used to fix remote ids of classes.
- `fixezurlobjectlinks.php`: is used to fix older occurrences of link items not being present in the `ezurl_object_table` for all versions/translations.
- `fixobjectremoteid.php`: is used to fix non-unique usage of content object remote ID's.
- `initurlaliasmlid.php`: initialize the `ezurlalias_ml_incr` table, part of the fixed issue #14077: eZURLAliasML database table lock and unlock code causes implicit commit of database transaction.
- `correctxmlalign.php`: after upgrading to 4.1 the default alignment for embedded images in `ezxmltext` datatype is removed, this script will allow you to have the correct `xmlalignment`.

Step 7: Clearing the caches

Whenever an eZ Publish solution is upgraded, all caches must be cleared in a proper way. This should be done from within a system shell:

1. Navigate into the eZ Publish 4.1 directory.
2. Run the script using the following shell command:

```
php bin/php/ezcache.php --clear-all --purge
```

Purging ensures that the caches are physically removed. When the "--purge" parameter is not specified, the caches will be expired but not removed.

Sometimes the script is unable to clear all cache files because of restrictive file/directory permission settings. Make sure that all cache files have been cleared by inspecting the contents of the various cache subdirectories within the "var" directory (typically the "var/cache/" and "var/<name_of_siteaccess>/cache/" directories). If there are any cache files left, you need to remove them manually.

Upgrading eZ Flow and/or Website Interface separately

Step 1: Backup

Before you will start the Website Interface or eZ Flow upgrading process make sure that you have backup of existing website state including database, extensions, INI settings, etc. During upgrade process existing ezflow and ezwebin extensions will be removed and replaced with new version. Website Interface and eZ Flow default content classes will be replaced as well. Upgrade script will also change some of the existing INI settings and add new ones which are required by latest version.

Step 2: eZ Flow upgrade

eZ Publish 4.1 comes with eZ Flow upgrade script which allows upgrade from previous version to the latest 1.1-0 version. To run upgrade process in automatic mode simply execute following command from eZ Publish root folder.

```
php bin/php/ezflowupgrade.php --to-version=1.1-0 --url=http://packages.ez.no/ezpublish/4.1/4.1.0
```

It is very important that user which will execute the upgrade script had correct permissions to modify content user extension, settings and var folders.

In order to get more information about available upgrade options, execute ezflowupgrade.php script with help param.

```
php bin/php/ezflowupgrade.php --help
```

Step 3: Re-write rules

Users running their eZ Flow sites in VH mode should update their extension related re-write rules to the following:

```
Rewriterule ^/extension/[^/]+/design/[^/]+/  
(stylesheets|flash|images|lib|jascripts?)/.* - [L]
```

Step 4: Website Interface upgrade

Upgrading Website Interface is very similar to eZ Flow upgrade process. In order to upgrade existing installation to latest 1.4-0 version simply execute following command from eZ Publish root folder:

```
php bin/php/ezwebinupgrade.php --to-version=1.4-0 --url=http://packages.ez.no/  
ezpublish/4.1/4.1.0
```

Again, it is very important that user which will execute the upgrade script had correct permissions to modify content user extension, settings and var folders.

In order to get more information about available upgrade options, execute ezwebinupgrade.php script with help param.

```
php bin/php/ezwebinupgrade.php --help
```

Step 5: Regenerate the autoload array for extensions

The autoload system has also some changes, for example the autoload array for extensions is now placed in var/autoload of your eZ Publish installation (along teh class changes in extensions itself).

To regenerate the autoload array, execute the following script from

```
php bin/php/ezpgenerateautoloads.php --extension
```

Step 6: Cache

Once the upgrade script is done, you need to clear the cache by executing following command from eZ Publish root folder:

```
php bin/php/ezcache.php --clear-all --purge
```

1.14 Upgrading to 4.0

This chapter contains information related to the upgrade of an eZ Publish solution to [version 4.0](#). Follow the upgrade procedures that are described in the following sections:

- [Important notes \(page 197\)](#)
- [Upgrading from 3.10.x to 4.0.y \(page 212\)](#)
- [Upgrading from 4.0.x to 4.0.y \(page 200\)](#)

1.14.1 Important notes

This section contains important notes about upgrading your existing eZ Publish 3.x installation to version 4.0. Read it carefully before you do the actual upgrade.

Upgrade path, requirements and preparation

eZ Publish 4.0 contains all the functionality of the 3.10 series. Before upgrading, check the requirements and set up the hosting environment for your site accordingly, so that you can quickly switch to PHP 5 after upgrading the distribution files.

It is not possible to upgrade directly from version 3.9 (and earlier) to 4.0, and the 4.0 distribution does not contain upgrade scripts for this. Instead, you should upgrade to 3.10.1 first, and then upgrade from 3.10.1 to 4.0.1.

Compatibility with extensions

Extensions compatible with eZ Publish 3.x may not work with 4.0. The following table shows which versions of the different extensions are supposed to work with eZ Publish 4.

Name	Compatibility with 4.0
Website Interface	v.1.3
Online Editor	v.4.3 (comes together with eZ Publish 4.0)
ODF Import / Export	v.2.1 (comes together with eZ Publish 4.0)
eZ PayPal Payment Gateway	v.1.1 (upcoming)
eZ Publish Extension for Oracle Database	v.1.8
eZ Find	v.1.0
eZ Newsletter	v.1.6 (upcoming)
Paynet Direct	Not tested for compatibility yet.
Paynet Terminal	Not tested for compatibility yet.

Custom extensions developed for eZ Publish 3.x may stop working after upgrading to 4.0, especially those using constants of the eZ Publish kernel. These have been moved inside class definitions. The following examples show the difference based on the code taken from "kernel/classes/ezcontentobject.php":

3.10:

```
...
define( "EZ_CONTENT_OBJECT_STATUS_DRAFT", 0 );
define( "EZ_CONTENT_OBJECT_STATUS_PUBLISHED", 1 );
define( "EZ_CONTENT_OBJECT_STATUS_ARCHIVED", 2 );
...
```

4.0:

```
class eZContentObject extends eZPersistentObject
{
    const STATUS_DRAFT = 0;
    const STATUS_PUBLISHED = 1;
    const STATUS_ARCHIVED = 2;
    ...
}
```

The following line shows an example of using these constants in PHP code for eZ Publish 3.x.

```
$status = EZ_CONTENT_OBJECT_STATUS_DRAFT;
```

In 4.0, this constant does not exist anymore (it has been renamed and moved inside the class definition) and you will have to use the new class constant instead:

```
$status = eZContentObject::STATUS_DRAFT;
```

Refer to the [PHP documentation](#) for more information about class constants.

Notes for extension developers

When developing custom extensions for eZ Publish 4, keep in mind that it stores each class definition in a separate PHP source file and makes use of the `_autoload()` function instead of having a list of needed includes at the beginning of each source file. All class definitions of the eZ Publish kernel have their paths listed in the "autoload/ezp_kernel.php" file. The "autoload/ezp_extension.php" file contains an array of paths for class definitions that are a part of the extensions that come with eZ Publish. This array must be updated every time you add a new extension to the system. This can be done by running the "ezpgenerateautoloads.php" script located in the "bin/php" directory. The following example shows how to run the script.

1. Navigate into the eZ Publish 4.0 directory.
2. Run the script using the following shell command:

```
php bin/php/ezpgenerateautoloads.php --extension
```

The script will look for class definitions in the "extension" directory and update the "autoload/ezp_extension.php" file accordingly. You can instruct the script to skip some of the extension subdirectories by using the optional "--exclude" parameter. To do this, run the script using the following shell command (replace "ext1" and "ext2" with the actual names of the extension subdirectories):

```
php bin/php/ezpgenerateautoloads.php --extension --exclude="extension/
ext1 extension/ext2"
```

Note that it is also possible to update the "autoload/ezp_extension.php" file by clicking the "Regenerate autoload arrays for extensions" button located in the "Setup - Extensions" part of the administration interface. In this case, only class definitions that are a part of globally enabled/activated extensions will be listed in the updated file.

1.14.2 from 4.0.x to 4.0.y

This section describes how to upgrade your existing eZ Publish 4.0.x installation to version 4.0.y. If you are upgrading from a version prior to eZ Publish 4.0.0, refer to the "Upgrading from 3.10.x to 4.0.y (page 212)" section for upgrade instructions.

In order to benefit from the latest bug fixes, you should upgrade directly to the latest stable release in the 4.0 branch (4.0.1 at the time of writing). Refer to the [changelogs](#) and [security advisories](#) for more information about the issues that were fixed in the latest releases.

Make sure that you have a working backup of the site before you do the actual upgrade. The upgrade procedure consists of the following steps:

1. Upgrading the distribution files to 4.0.1
2. Upgrading the database to 4.0.1
3. Running the system upgrade scripts
4. Updating the system configuration
5. Clearing the caches

Step 1: Upgrading the distribution files

The easiest way to upgrade the distribution files is to unpack eZ Publish 4.0.1 to a separate directory and then copy the directories that contain site-specific files from the existing installation. Make sure that you copy the following directories:

- design/example
- design/example_admin
- var
- settings/siteaccess
- settings/override

Replace "example" and "example_admin" with the actual names of your siteaccesses.

Custom extensions

If you are using custom extensions, the subdirectories inside the "extension" directory will also have to be copied. However, make sure that you do not overwrite any extensions that come with eZ Publish (currently "ezdhtml", "ezodf" and "ezurlaliasmigration"). Note that upgrading the distribution files will overwrite the autoload arrays for extensions. You will need to re-generate

the autoload arrays for active extensions later (after enabling the "ezurlaliasmigration" extension in step 3).

If you are using the Website Interface front-end, be sure to copy the "extension/ezwebin/" directory.

Important bug fix for remote IDs

Due to some bugs in previous versions of eZ Publish (3.9.0 and others), it might happen that not all content objects in your database have unique remote IDs. This basically means that the "remote_id" column of the "ezcontentobject" table in the database might contain duplicate values. In eZ Publish 4.0.1, a database schema change was introduced in order to ensure that only unique remote IDs can be stored in this table. Because of this, you need to make sure that there are no content objects with identical remote IDs before upgrading the database. This can be done by running the "fixobjectremoteid.php" upgrade script. (Note that both the database schema change and the upgrade script were added to eZ Publish 3.9.5 and 3.10.1 as well.)

This script should be run once for each database, specifying one siteaccess per database. If you only have a public and an administration siteaccess that share the same database (which is the most typical/usual case), you only need to run the script for one of the siteaccesses. If the siteaccess isn't specified, the default siteaccess will be used.

The following example shows how to run the script:

1. Navigate into the eZ Publish 4.0.1 directory.
2. Run the script (replace "example" with the actual name of your siteaccess):

```
php update/common/scripts/4.0/fixobjectremoteid.php -s example
```

The script will search for objects with non-unique remote IDs. Every time such an object is found, the script will suggest to either display more detailed information or fix the problem automatically. If you choose the first option, the script will display the list of objects that have the same remote ID and let you decide which one should remain unchanged; all other objects in the list will get new remote IDs. Otherwise, the script will fix the problem automatically based on the objects' creation dates. This means that the script will generate new remote IDs for all objects in the list except the one that was created first.

The optional "--mode" parameter makes it possible to run the script in either "auto-fix" or "manual-fix" mode as shown below.

```
php update/common/scripts/4.0/fixobjectremoteid.php -s example --mode=value
```

If you replace "value" with "a" in the command above, the script will automatically fix all non-unique remote IDs that are found in the database based on the objects' creation dates. In this case, no further input is required from the user. If you specify "--mode=d", the script will display the list of objects for each non-unique remote ID so that you can manually choose which objects should keep their remote IDs.

Step 2: Upgrading the database

Follow the instructions below to upgrade your eZ Publish database from version 4.0.0 to 4.0.1.

Switching to InnoDB (MySQL only)

If you are using MySQL, note that the recommended storage engine is InnoDB. This storage engine is used by default during the installation process when the setup wizard initializes the database structure for eZ Publish. In addition, InnoDB will most likely be required for future versions of eZ Publish to run on MySQL. Contact your database administrator if you are unsure about whether InnoDB is available on your server.

In the future, all database upgrade scripts will use the InnoDB storage engine when creating new tables. Because of this, it is recommended (but not required) to make sure that all existing tables in your database use InnoDB. If you want to use the MyISAM (or another) storage engine instead, all existing tables in your database must use this engine.

You can use the "bin/php/ezconvertmysqltabletype.php" script to find out about the current storage engine in your database and convert your tables to use another engine if needed (refer to the "Switching to InnoDB" part of the "Upgrading from 3.10.x to 4.0.y" documentation page for more information about this script).

Running the database upgrade scripts

The following text describes how to run the database upgrade scripts. (Note that even though the sql file contains a section marked with "from 3.10.1" comments, you do not need to skip this part when upgrading from 4.0.0 to 4.0.1.)

MySQL

1. Navigate into the eZ Publish 4.0.1 directory.
2. Run the database upgrade script:

```
mysql -u <username> -p<password> <database> < update/database/mysql/4.0/  
dbupdate-4.0.0-to-4.0.1.sql
```

PostgreSQL

1. Navigate into the eZ Publish 4.0.1 directory.
2. Run the database upgrade script:

```
psql -d <database> -U <dbowner> < update/database/postgresql/4.0/  
dbupdate-4.0.0-to-4.0.1.sql
```

Specific steps for upgrading to versions 4.0.2 until 4.0.7 are located here.

Step 3: Running the system upgrade scripts

The 4.0.1 version of eZ Publish introduces a couple of important bug fixes and functionality changes. In order to make sure that your site is compatible with these changes, you may need to run a few upgrade scripts.

Multi-language support for URL aliases

In eZ Publish 4.0.1, the multi-language support for URL aliases was substantially improved, along with some important bug fixes related to this functionality. When upgrading, you need to regenerate all URL aliases that are stored in the "ezurlalias_ml" database table. This basically means that all auto-generated URL aliases will be deleted and then created from scratch by using the "updateniceurls.php" script. However, manual/user-defined URL aliases and URL history entries cannot be regenerated automatically. Instead, these items will be transferred/migrated into a temporary table in the database (hereinafter called "migration table") and then restored at a later stage. This must be done by using the "migrate.php" script, which is included in the "ezurlalias-migration" extension. This new extension is included in eZ Publish 4.0.1 and provides tools for the purpose of URL alias migration.

Follow the instructions below in order to update your URL aliases.

1. Create a database backup.

Before you continue, create a backup of your database (or at least make sure that you have a backup of the "ezurlalias_ml" table itself).

2. Enable the "ezurlaliasmigration" extension.

The "ezurlaliasmigration" extension must be enabled for all of your siteaccesses. To do this, edit the "site.ini.append.php" file located in the "settings/override" directory and add the following line under the [ExtensionSettings] section:

```
ActiveExtensions []=ezurlaliasmigration
```

Alternatively, you can enable the extension separately for each siteaccess. To do this, edit the "site.ini.append.php" file located in the "settings/siteaccess/example" directory (replace "example" with the name of the target siteaccess) and add the following line under the [ExtensionSettings] section:

```
ActiveAccessExtensions []=ezurlaliasmigration
```

3. Update the autoload arrays for extensions (optional).

If you are using custom extensions, some of them might stop working after upgrading to 4.0.1. The reason for this is that the "autoload/ezp_extension.php" file is overwritten in the first step of the upgrade procedure (when upgrading the distribution files). To solve the problem, you need to regenerate the autoload arrays for active extensions as described below.

Note that the default "autoload/ezp_extension.php" file in eZ Publish 4.0.1 contains all the information needed for the following extensions to work correctly: "ezdhtml", "ezodf", "ezurlaliasmigration", "ezflow" and "ezwebin". If you are only using these extensions (no custom ones), you do not have to regenerate the autoload arrays when upgrading to 4.0.1.

The following example shows how to regenerate the autoload arrays for extensions.

1. Navigate into the eZ Publish 4.0.1 directory.
2. Run the script using the following shell command:

```
php bin/php/ezpgenerateautoloads.php --extension
```

The script will look for class definitions in the "extension" directory and update the "autoload/ezp_extension.php" file accordingly.

Note that the command above makes the script go through all extensions regardless of whether or not they are enabled. You can also instruct the script to skip inactive extensions by listing them in the optional "--exclude" parameter (replace "ext1" and "ext2" with the actual names of the extension subdirectories):

```
php bin/php/ezpgenerateautoloads.php --extension --exclude="extension/  
ext1 extension/ext2"
```

4. Create the migration table.

The "ezurlaliasmigration" extension requires a custom database table called "ezurlalias_migration" to be created in the database. This table will be used for storing the migrated URL alias data. The new table must be created in accordance with the definitions specified in the "schema.sql" file included in the extension. This can be done by executing the SQL queries from the "schema.sql" file as described below.

If you are using MySQL, navigate into the eZ Publish 4.0.1 directory and run the following shell command:

```
mysql -u <username> -p<password> <database> < extension/ezurlaliasmigration/  
sql/mysql/schema.sql
```

If you are running PostgreSQL, use the following shell command instead:

```
psql -d <database> -U <dbowner> < extension/ezurlaliasmigration/sql/postgresql/
schema.sql
```

Alternatively, you can use the "migrate.php" script to create the migration table. This option will work regardless of which database implementation you are using. You need to navigate into the eZ Publish 4.0.1 directory and run the script from within the system shell:

```
php extension/ezurlaliasmigration/scripts/migrate.php --create-migration-table
```

With this command, the script will create the "ezurlalias_ml_migrate" table in the database that is used by the default siteaccess. If your site makes use of two or more databases, you need to run the script once for each database, specifying one siteaccess per database. Use the "-s" parameter to specify the target siteaccess as shown below:

```
php extension/ezurlaliasmigration/scripts/
migrate.php -s example --create-migration-table
```

(Replace "example" with the actual name of your siteaccess.)

5. Transfer your URL aliases to the migration table.

To find out about the current number of manual/user-defined URL aliases and URL history entries in your system, run the "migrate.php" script without parameters or with the "-s" parameter only. The following example shows how this can be done.

1. Navigate into the eZ Publish 4.0.1 directory.
2. Run the script (replace "example" with the actual name of your siteaccess):

```
php extension/ezurlaliasmigration/scripts/migrate.php -s example
```

Note that the commands for running the "migrate.php" and "updateniceurls.php" scripts in the examples below will assume that all your siteaccesses share the same database. Because of this, the "-s example" part will be omitted and the scripts will be run for the database that is used by the default siteaccess. If you are using two or more databases, the "migrate.php" script should be run once for each database, specifying one siteaccess per database (by adding the "-s" parameter to each command). This also applies to the "updateniceurls.php" script. Both scripts need to be run from the root directory of your eZ Publish 4.0.1 installation. The following table reveals the available parameters for the "migrate.php" script.

Parameter	Description
--create-migration-table	Create the migration table in the database.
--migrate-alias	Transfer manual URL aliases to the migration table.
--migrate-history	Transfer URL history entries to the migration

	table.
--migrate	Transfer both manual URL aliases and URL history entries to the migration table.
--restore-alias	Restore manual URL aliases from the migration table.
--restore-history	Restore URL history entries from the migration table.
--restore	Restore both manual URL aliases and URL history entries from the migration table.

To transfer your manual URL aliases and URL history entries into the migration table, run the "migrate.php" script as shown below:

```
php extension/ezurlaliasmigration/scripts/migrate.php --migrate
```

The script will go through all manual URL aliases and URL history entries in the "ezurlalias_ml" table and put the collected data into the "ezurlalias_ml_migrate" table.

Instead of transferring both manual URL aliases and URL history entries at once, it is also possible to run the script once for each type of URL entries. To do this, use either the "--migrate-alias" or "--migrate-history" parameter instead of "--migrate" when running the script (refer to the table above for more information about these parameters).

It is important not to change any configuration settings after transferring the data to the migration table. The configuration of the URL alias system must remain untouched until you're finished with regenerating the URL aliases and transferring the data back to the "ezurlalias_ml" table (otherwise, there might be some broken or inconsistent URLs on your site after upgrading to 4.0.1).

6. Remove old URL alias data.

Use the following SQL query to empty the "ezurlalias_ml" table in your database:

```
TRUNCATE ezurlalias_ml;
```

7. Create new auto-generated URL aliases.

Now that the "ezurlalias_ml" table is empty, you need to create new auto-generated URL aliases based on the existing node tree structure and content object names. When you do this, it is recommended that you clear ini cache first, in order to avoid using old settings from cache (This can be done in the administration interface). Then run the "updateniceurls.php" script using the following command:

```
php bin/php/updateniceurls.php --fetch-limit=number
```

Specify the desired number of items to handle per one iteration instead of "number". On a system with a 128 MB memory limit for PHP, this parameter can be set to 2000. (The script

will automatically do as many iterations as necessary to generate the URL aliases for all nodes, based on the number specified. If the optional "--fetch-limit" parameter is omitted, the script will handle 200 items per one iteration.)

The script will go through all content objects stored in the database and create new virtual URLs for them in accordance with the current transformation settings. New URL aliases will be generated for all of the translation languages and placed in the "ezurlalias_ml" table in the database.

8. Restore migrated URL data.

You need to restore the custom URL aliases and URL history entries from the migration table. This can be done by running the "migrate.php" script:

```
php extension/ezurlaliasmigration/scripts/migrate.php --restore
```

The script will go through all entries in the "ezurlalias_ml_migrate" table and put the collected data back into the "ezurlalias_ml" table.

Instead of restoring both manual URL aliases and URL history entries at once, it is also possible to run the script once for each type of URL entries. To do this, use either the "--restore-alias" or "--restore-history" parameter instead of "--restore" when running the script (refer to the table above for more information about these parameters).

It might happen that a URL alias entry was previously corrupted and cannot be automatically restored from the migration table. In this case, you will see "F" in the console output of the "migrate.php" script. The following SQL query can be used in order to display the migrated entries that were not restored:

```
SELECT id, link, parent, lang_mask, text, action, is_original, is_alias,  
lang_mask_adjusted FROM ezurlalias_ml_migrate WHERE is_restored=0;
```

If you want the URL entries that were not restored automatically to work on your site, you must create new user-defined URLs later (when the upgrade procedure is finished). This can be done from within the administration interface as described on the "Managing URL aliases" documentation page.

Specific steps regarding URL alias upgrade when upgrading to version 4.0.2 is located here.

Changes to system upgrade scripts (optional)

If your solution was initially built on eZ Publish 3.10 or 4.0, skip this part. Otherwise, read it carefully.

The "updatevatcountries.php" upgrade script in eZ Publish 3.10.0 contains a bug. The same bug exists in eZ Publish 3.9.3 and 3.9.4 (refer to <http://issues.ez.no/11955> for more information).

The updated version of the script is available in eZ Publish 3.9.5, 3.10.1 and 4.0.1. If you previously upgraded to 3.10.0 using the original version of the script and then didn't run the updated version when upgrading from 3.10.0 to 4.0.0, you need to run it when upgrading to 4.0.1.

This script should be run once for each database, specifying one siteaccess per database. If you only have a public and an administration siteaccess that share the same database (which is the most typical/usual case), you only need to run the script for one of the siteaccesses. If the siteaccess isn't specified, the default siteaccess will be used.

The following example shows how to run the script:

1. Navigate into the eZ Publish 4.0.1 directory.
2. Run the script (replace "example" with the actual name of your siteaccess):

```
php update/common/scripts/4.0/updateevatcountries.php -s example
```

The script will go through all the VAT rules in the database and update them.

Step 4: Updating the system configuration

You are not required to do any configuration changes when upgrading from 4.0.x to 4.0.y.

Tree menu configuration (optional)

eZ Publish 4.0.1 introduces some changes to the dynamic tree menu functionality. In particular, the "treemenu" view of the "content" module no longer makes use of parameters that are transferred using GET variables (this improves tree menu compatibility when running PHP in [FastCGI](#) mode; refer to <http://issues.ez.no/11806> for more information).

If you have instructed Apache to use "index_treemenu.php" instead of "index.php" when the treemenu view of the content module is requested, this feature will stop working after upgrading to version 4.0.1. To make it work, update your rewrite rules as described below.

If your ".htaccess" file contains the following line:

```
RewriteRule content/treemenu/?$ index_treemenu.php
```

you need to remove "\$" from there. The updated line should look like this:

```
RewriteRule content/treemenu/? index_treemenu.php
```

If no ".htaccess" file is used, check whether the following line is present in your Apache configuration file:

```
RewriteRule content/treemenu/?$ /index_treemenu.php [L]
```

If found, remove "\$" from this line. The result will look like this:

```
RewriteRule content/treemenu/? /index_treemenu.php [L]
```

Note that due to a bug in eZ Publish 4.0.1, you need to download the updated version of "index_treemenu.php" from <http://pubsvn.ez.no> and replace the corresponding file in your installation. The same [bug](#) exists in eZ Publish 3.10.1.

Step 5: Clearing the caches

Whenever an eZ Publish solution is upgraded, all caches must be cleared in a proper way. This should be done from within a system shell:

1. Navigate into the eZ Publish 4.0.1 directory.
2. Run the script using the following shell command:

```
php bin/php/ezcache.php --clear-all --purge
```

Purging ensures that the caches are physically removed. When the "--purge" parameter is not specified, the caches will be expired but not removed (refer to this page for more information).

Sometimes the script is unable to clear all cache files because of restrictive file/directory permission settings. Make sure that all cache files have been cleared by inspecting the contents of the various cache subdirectories within the "var" directory (typically the "var/cache/" and "var/<name_of_siteaccess>/cache/" directories). If there are any cache files left, you need to remove them manually.

Upgrading to 4.0.2

URL Alias upgrade

The 4.0.2 release expands on the work which was made to the URL alias system in 4.0.1. The fix to the URL alias system, changes the structure in which URL entries are stored. This means that the URL alias entries needs to be regenerated, as was the case when upgrading to 4.0.1.

Follow the above upgrade steps to upgrade the URL aliases. With one important difference:

In the last round of fixes made in this area, we discovered that it is unfortunately not possible to migrate the URL alias history entries. This is because they cannot be restored properly. This means that in the migration process, the *migrate.php* script should be run with the options to migrate aliases only (custom aliases are retained, as well as all future history elements).

A simplified run down of the process below. Please advise with the full upgrade documentation for more detail.

- Migrate custom URL aliases

```
php extension/ezurlaliasmigration/scripts/migrate.php --migrate-alias
```

- To clean up existing URL alias table, run this query:

```
truncate ezurlalias_ml;
```

- Regenerate URL alias entries for the site access *example*, replace with your own value

```
bin/php/updateniceurls.php -s example --fetch-limit=number
```

- Restore custom URL aliases

```
php extension/ezurlaliasmigration/scripts/migrate.php --restore-alias
```

General upgrade scripts

First the database update queries need to be run. Then the upgrade scripts listed below, must be run. Use the **-h** parameter to see usage on how to run them.

- `php update/common/scripts/4.0/initurlaliasmlid.php`
- `php update/common/scripts/4.0/fixezurlobjectlinks.php`
- `php update/common/scripts/4.0/fixclassremoteid.php`

Upgrading to 4.0.3

If you are not upgrading from eZ Publish 4.0.2, first make sure that the upgrade steps for 4.0.2 are performed, as listed above. For upgrading from eZ Publish 4.0.2 to eZ Publish 4.0.3, only the database queries in *dbupdate-4.0.2-to-4.0.3.sql* are required.

Upgrading to 4.0.4

If you are not upgrading from eZ Publish 4.0.3, first make sure that the upgrade steps for 4.0.3 are performed, as listed above. For upgrading from eZ Publish 4.0.3 to eZ Publish 4.0.4, only the database queries in *dbupdate-4.0.3-to-4.0.4.sql* are required.

Upgrading to 4.0.5

If you are not upgrading from eZ Publish 4.0.4, first make sure that the upgrade steps for 4.0.4 are performed, as listed above. For upgrading from eZ Publish 4.0.4 to eZ Publish 4.0.5, only the database queries in *dbupdate-4.0.4-to-4.0.5.sql* are required.

Upgrading to 4.0.6

If you are not upgrading from eZ Publish 4.0.5, first make sure that the upgrade steps for 4.0.5 are performed, as listed above. For upgrading from eZ Publish 4.0.5 to eZ Publish 4.0.6, only the database queries in *dbupdate-4.0.5-to-4.0.6.sql* are required.

Upgrading to 4.0.7

If you are not upgrading from eZ Publish 4.0.6, first make sure that the upgrade steps for 4.0.6 are performed, as listed above. For upgrading from eZ Publish 4.0.6 to eZ Publish 4.0.7, only the database queries in *dbupdate-4.0.6-to-4.0.7.sql* are required.

1.14.3 from 3.10.x to 4.0.y

This section describes how to upgrade your existing eZ Publish 3.10.x installation to version 4.0.y. If you are upgrading from version 3.10.0 or earlier, you need to first upgrade to 3.10.1 before you can upgrade to 4.0. (Refer to either "Upgrading from 3.9.x to 3.10.y (page 242)" or "Upgrading from 3.10.x to 3.10.y (page 235)" depending on which version you are currently running.)

In order to benefit from the latest bug fixes, you should upgrade directly to the latest stable release in the 4.0 branch (4.0.1 at the time of writing). Refer to the [changelogs](#) and [security advisories](#) for more information about the issues that were fixed in the latest releases. Note that there is a significant problem with eZ Publish 4.0.0. More precisely, the "ezsmtp" class contains a [bug](#) that prevents the system from sending outgoing mails through an SMTP server. This bug was fixed in version 4.0.1.

Note that in some cases, it is not possible or not recommended to upgrade from 3.10 to 4.0. Read the "Important notes (page 197)" section in order to find out whether you can upgrade immediately or have to delay upgrading until new software releases are available.

The upgrade procedure described below is generic and does not cover any specific cases (for example, running eZ Publish in a clustered environment).

Make sure that you have a working backup of the site before you do the actual upgrade.

The procedure for upgrading directly from version 3.10.x to 4.0.y consists of the following steps:

1. Upgrading the distribution files to 4.0.y
2. Upgrading the database to 4.0.y
3. Running the system upgrade scripts for versions from 3.10.x to 4.0.y
4. Updating the system configuration
5. Clearing the caches

Step 1: Upgrading the distribution files

The easiest way to upgrade the distribution files is to unpack eZ Publish 4.0.y to a separate directory and then copy the directories that contain site-specific files from the existing installation. Make sure that you copy the following directories:

- design/example
- design/example_admin
- var
- settings/siteaccess
- settings/override

Replace "example" and "example_admin" with the actual names of your siteaccesses.

Custom extensions

If you are using custom extensions, the subdirectories inside the "extension" directory will also have to be copied. However, make sure that you do not overwrite any extensions that come with eZ Publish (currently "ezdhtml", "ezodf" and "ezurlaliasmigration").

If you are using the Website Interface front-end, you will need to upgrade the extension to version 1.3 as earlier versions are not compatible with eZ Publish 4. Refer to the "Upgrading the Website Interface" chapter of the Website Interface installation guide for more information and instructions.

Switching to PHP 5

After upgrading the distribution files, switch your site to use PHP 5 (make sure all the requirements are met).

Note that you need to set the "date.timezone" value in the "php.ini" configuration file. If this setting is not specified, you will most likely receive error messages like "It is not safe to rely on the system's timezone settings" when running eZ Publish on PHP 5. The following example shows how the corresponding line in "php.ini" looks like:

```
date.timezone = <timezone>
```

Refer to the [PHP documentation](#) for the list of supported timezones. Don't forget to restart Apache after editing "php.ini".

Step 2: Upgrading the database

The following text describes how to upgrade your eZ Publish database from version 3.10.x to 4.0.y.

Switching to UTF-8

eZ Publish 4 requires UTF-8 as the character set. If your site makes use of another character set, you must switch to UTF-8. To find out about the current character set in your database, you can run the following command using the [MySQL command-line tool](#) (replace "db_name" with the actual name of your database):

```
mysql> SHOW CREATE DATABASE db_name;
```

PostgreSQL users can view the list of available databases along with their character sets by running the following command from within the system shell:

```
psql -l
```

In eZ Publish 4.0.1, the "bin/php/ezconvertdbcharset.php" script was added in order to convert databases to use UTF-8. (Refer to the [developer documentation](#) for more information and troubleshooting tips).

This script should be run once for each database, specifying one siteaccess per database. If you only have a public and an administration siteaccess that share the same database (which is the most typical/usual case), you only need to run the script for one of the siteaccesses. If the siteaccess isn't specified, the default siteaccess will be used.

Before running the script, make sure that eZ Publish knows the current character set in the database. Basically, the script will rely on the "Charset" setting located in the "[DatabaseSettings]" section of "settings/override/site.ini.append.php" or "settings/siteaccess/<your_siteaccess>/site.ini.append.php". If this setting is empty, the value of the "Charset" setting located in the [CharacterSettings] section of "i18n.ini" (or an override) will be used.

The following example shows how to run the script:

1. Navigate into the eZ Publish 4.0.1 directory.
2. Run the script (replace "example" with the actual name of your siteaccess):

```
php bin/php/ezconvertdbcharset.php -s example
```

The script will convert the database to use UTF-8 as the character set. After running the script, make sure that eZ Publish knows about the database's new character, and also update the eZ Publish internal character set accordingly, in the correct override files:

- File: "site.ini":

```
[DatabaseSettings]
Charset=utf-8
```

- File: "i18n.ini":

```
[CharacterSettings]
Charset=utf-8
```

Switching to InnoDB (MySQL only)

If you are using MySQL, note that the recommended storage engine is InnoDB. This storage engine is used by default during the installation process when the setup wizard initializes the database structure for eZ Publish. In addition, InnoDB will most likely be required for future

versions of eZ Publish to run on MySQL. Contact your database administrator if you are unsure about whether InnoDB is available on your server.

In eZ Publish 4.0.1, the database upgrade scripts use the InnoDB storage engine when creating new tables. Before running these scripts, it is recommended (but not required) to make sure that all existing tables in your database use InnoDB. If you want to use the MyISAM (or another) storage engine instead, all existing tables in your database must use this engine. In addition, you will have to replace "InnoDB" with "MyISAM" in the database upgrade scripts before running them. Read the following text for more detailed instructions.

To find out about the current storage engine in your database, navigate into the eZ Publish directory and run the "bin/php/ezconvertmysqltabletype.php" script with the "--list" parameter:

```
php bin/php/ezconvertmysqltabletype.php --list
```

The script will output the list of tables in the database along with the name of the storage engine for each table.

With the command above, the script will check the table types in the database that is used by the default siteaccess. If your site makes use of two or more databases, you need to run the script once for each database, specifying one siteaccess per database. Use the "-s" parameter to specify the target siteaccess as shown below (replace "example" with the actual name of your siteaccess):

```
php bin/php/ezconvertmysqltabletype.php --list -s example
```

If all tables in your eZ Publish database use the InnoDB storage engine, no changes are needed. If some tables use MyISAM or another storage engine, it is recommended (but not required) to convert them to InnoDB. To do this, run the script with the "--newtype" parameter as shown below:

```
php bin/php/ezconvertmysqltabletype.php --newtype=innodb -s example
```

The script will convert all tables in your database to InnoDB. Note that the script will automatically skip the tables that already use the InnoDB storage engine from before (if any).

You can also convert tables individually by using the following SQL query for each table:

```
ALTER TABLE <name_of_table> TYPE = innodb;
```

If you wish to use a storage engine other than InnoDB, make sure that all tables in your database use this engine. In addition, you will have to change the engine specified in the first line of the database upgrade scripts before running them (otherwise, the upgrade will leave you with a mix of table types).

You will find the database upgrade scripts in the "update/database/mysql/4.0/" directory. Each database upgrade script that creates new tables contains the following line in the beginning:

```
SET storage_engine=INNODB;
```

Before you run such a script, replace "INNODB" with the name of your storage engine. For example, if all existing tables in your database use MyISAM, you need to modify the first line as shown below:

```
SET storage_engine=MYISAM;
```

This will make the database upgrade script use MyISAM instead of InnoDB when creating new tables.

Running the database upgrade scripts

To upgrade your database from version 3.10.1 to 4.0.1, you need to navigate into the eZ Publish 4.0.1 directory and run the following database upgrade scripts one after another:

1. dbupdate-3.10.0-to-4.0.0.sql
2. dbupdate-4.0.0-to-4.0.1.sql

Note that some of the database changes from eZ Publish 3.10.0 to 4.0.1 have been included in the 3.10.1 release as well. This means that you need to skip some parts of the database upgrade scripts when upgrading from version 3.10.1 to 4.0.1. These are marked with "from 3.10.1" comments in the ".sql" files.

MySQL

The database upgrade scripts are located in the "update/database/mysql/4.0/" directory of your eZ Publish installation. Each of these scripts can be launched using the following shell command (replace "script_name" with the actual name of the script):

```
mysql -u <username> -p<password> <database> <  
update/database/mysql/4.0/script_name.sql
```

PostgreSQL

The database upgrade scripts are located in the "update/database/postgresql/4.0/" directory of your eZ Publish installation. Each of these scripts can be launched using the following shell command (replace "script_name" with the actual name of the script):

```
psql -d <database> -U <dbowner> < update/database/postgresql/4.0/script_name.sql
```

Step 3: Running the system upgrade scripts

You are not required to run any system upgrade scripts when upgrading from 3.10.1 to 4.0.1.

Step 4: Updating the system configuration

In eZ Publish versions prior to 4.0.0, the order of design resources used by the "ezdesign" and "ezimage" template operators was not the same as used by the "include" template function and eZTemplate::fetch(). This might lead to problems with overriding templates in extensions (refer to <http://issues.ez.no/9595> for more information). This issue was fixed in eZ Publish 4.0.0.

In eZ Publish 4, templates that override default ones must be placed in the "override/templates" directory of your design. If you are not sure about the location of override templates in your eZ Publish installation, review the settings in your "override.ini.append.php" configuration file. (Refer to "The template override system" for more information.) The names of the override templates are specified using the "MatchFile" directive, which makes it possible to find the corresponding files on a filesystem.

Note that custom templates that are not connected with the override system must be placed in the "templates" directory of a custom design.

Step 5: Clearing the caches

Whenever an eZ Publish solution is upgraded, all caches must be cleared in a proper way. This should be done from within a system shell:

1. Navigate into the eZ Publish 4.0 directory.
2. Run the script using the following shell command:

```
php bin/php/ezcache.php --clear-all --purge
```

Purging ensures that the caches are physically removed. When the "--purge" parameter is not specified, the caches will be expired but not removed (refer to this page for more information).

Sometimes the script is unable to clear all cache files because of restrictive file/directory permission settings. Make sure that all cache files have been cleared by inspecting the contents of the various cache subdirectories within the "var" directory (typically the "var/cache/" and "var/<name_of_siteaccess>/cache/" directories). If there are any cache files left, you need to remove them manually.

1.14.4 from 3.10.0 to 4.0.0 (deprecated)

This documentation page is deprecated and no longer maintained. Refer to the following documentation pages for upgrade instructions:

- Upgrading from 3.10.x to 4.0.y (page [212](#))
- Upgrading from 4.0.x to 4.0.y (page [200](#))

This section describes how to upgrade your existing eZ Publish 3.10.0 installation to version 4.0.0. If you are upgrading from a version prior to eZ Publish 3.10.0, you need to first upgrade to 3.10.0 (page [254](#)) before you can upgrade to 4.0.0. In this case, there are two system upgrade scripts that you should skip running when upgrading from 3.9.x to 3.10.0:

- In eZ Publish 4.0, the "bin/php/updateniceurls.php" script was sufficiently improved. In addition, the support for wildcard based URL forwarding was re-added and the default type of URL transformation was changed to "urlalias". Refer to <http://issues.ez.no/11650> and <http://issues.ez.no/12093> for more information.
- The "updatetipafriendpolicy.php" upgrade script in eZ Publish 3.10.0 contains a bug. It prevents the script from granting access to the "Tip a friend" feature to users. The problem occurs when the script is run on Windows. This issue was fixed in 4.0.0 (plus the upcoming versions 3.10.1 and 3.9.5). Refer to <http://issues.ez.no/11663> for more information.

How to upgrade

Note that in some cases, it is not possible or not recommended to upgrade from version 3.10.0 to 4.0.0. Read the "Important notes (page [197](#))" section in order to find out whether you can upgrade immediately or have to delay upgrading until new software releases are available.

The upgrade procedure described below is generic and does not cover any specific cases (for example, running eZ Publish in a clustered environment).

eZ Publish 4.0 contains all the functionality of the 3.10 series. Before upgrading, check the requirements and set up the hosting environment for your site accordingly, so that you can quickly switch to PHP 5 after upgrading the distribution files.

Make sure that you have a working backup of the site before you do the actual upgrade. The upgrade procedure consists of the following steps:

1. Upgrading the distribution files to 4.0
2. Upgrading the database to 4.0
3. Running the 4.0 upgrade scripts
4. Updating the system configuration
5. Clearing the caches

Step 1: Upgrading the distribution files

The easiest way to upgrade the distribution files is to unpack eZ Publish 4.0 to a directory and then copy the directories that contain site-specific files from the existing installation. Make sure that you copy the following directories:

- design/example
- design/example_admin
- var
- settings/siteaccess
- settings/override

Replace "example" and "example_admin" with actual names used by your siteaccesses.

Custom extensions

If you are using custom extensions, the subdirectories inside the "extension" directory will also have to be copied. However, make sure that you do not overwrite any extensions that come with eZ Publish (currently "ezodf" and "ezdhtml").

If you are using the Website Interface front-end, you will need to upgrade the extension to version 1.3 as earlier versions are not compatible with eZ Publish 4. Refer to the "Upgrading the Website Interface" chapter of the Website Interface installation guide for more information and instructions.

Switching to PHP 5

After upgrading the distribution files, switch your site to use PHP 5 (make sure all the requirements are met).

Note that you need to set the "date.timezone" value in the "php.ini" configuration file. If this setting is not specified, you will most likely receive error messages like "It is not safe to rely on the system's timezone settings" when running eZ Publish on PHP 5. The following example shows how the corresponding line in "php.ini" looks like:

```
date.timezone = <timezone>
```

Refer to the [PHP documentation](#) for the list of supported timezones. Don't forget to restart Apache after editing "php.ini".

Step 2: Upgrading the database

The following text describes how a 3.10.0 database can be upgraded to 4.0.0.

MySQL

1. Navigate into the eZ Publish 4.0.0 directory.
2. Run the database upgrade script:

```
mysql -u <username> -p<password> <database> <  
update/database/mysql/4.0/dbupdate-3.10.0-to-4.0.0.sql
```

Note that the CREATE TABLE statements in the database upgrade script do not specify which storage engine to use (no ENGINE or TYPE option), and thus the default storage engine will be used. Normally, it is MyISAM. If you are using InnoDB, make sure the default storage engine is set to InnoDB before you run the database upgrade script (refer to the [MySQL documentation](#) for information about how to set the default engine). If you were not able to change the MySQL configuration on your server, and the upgrade left you with a mix of table types, you can convert the newly created table to InnoDB using the following SQL query:

```
ALTER TABLE ezurlwildcard TYPE = innodb;
```

It is also possible to use the "bin/php/ezconvertmysqltabletype.php" script for database conversion.

PostgreSQL

1. Navigate into the eZ Publish 4.0.0 directory.
2. Run the database upgrade script:

```
psql -d <database> -U <dbowner> < update/database/postgresql/4.0/  
dbupdate-3.10.0-to-4.0.0.sql
```

Step 3: Running the system upgrade scripts

Binary files uploaded via WebDAV (optional)

In eZ Publish version 3.9.0 - 3.9.4 and 3.10.0, there is a bug in the "File" datatype which may cause problems with binary file uploading via WebDAV. The typical symptoms are that the file extension of the uploaded file is missing and that you may not be able to download the file once it is uploaded (refer to <http://issues.ez.no/9450> for more information). This issue was fixed in eZ Publish 4.0.0.

If your site has objects that were created by uploading files via WebDAV and you have configured MIME-type-to-class mapping using the "MimeClassMap[]" directive in an override for "upload.ini", you need to run the "updatebinaryfile.php" script in order to fix the file extensions. (If a content class does not have any attributes that make use of the "File" datatype, instances

of this class are not affected.) This script should be run for all siteaccesses that use different databases. If you only have a public and an administration siteaccess (which is the most typical/usual case), you only need to run the script for one of the siteaccesses. If the siteaccess isn't specified, the default siteaccess will be used.

The following example shows how to run the script.

1. Navigate into the eZ Publish 4.0 directory.
2. Run the script (replace "example" with the actual name of your siteaccess):

```
php update/common/scripts/4.0/updatebinaryfile.php -s example
```

The script will go through all the binary files and fix their extensions. The filename entries stored in the "ezbinaryfile" database table will be changed accordingly.

Multi-language support for URL aliases (optional)

eZ Publish 4.0 supports wildcard based URL forwarding. This feature was removed when implementing the multilingual URLs functionality for eZ Publish 3.10.0 and then re-added in later versions. In eZ Publish 4.0, the upgrade script for generating multilingual URL aliases was sufficiently improved. In addition, the default type of URL transformation was changed to "urlalias" (refer to <http://issues.ez.no/12093> for more information).

If you are upgrading from 3.9.x to 4.0.0, you need to upgrade to 3.10.0 first and then upgrade to 4.0.0. In this case, it is recommended to skip running the "updateniceurls.php" script when upgrading to 3.10.0, and run the updated version of the script when upgrading from 3.10.0 to 4.0.0.

If you are running eZ Publish 3.10.0 and experiencing any problems with multilingual URL aliases, these can be a result of changes that were made by the "updateniceurls.php" script when upgrading to 3.10.0. In order to avoid such problems in the future, it is recommended to revert the changes and run the updated version of the script when upgrading from 3.10.0 to 4.0.0. However, this operation is not safe: in case you have added new custom URL aliases in 3.10.0, these will be lost.

Reverting the changes

In order to revert the changes that were made by the "updateniceurls.php" script when upgrading to 3.10.0, run the following SQL queries:

```
UPDATE ezurlalias SET is_imported=0;  
TRUNCATE ezurlalias_ml;
```

Running the upgrade script

After reverting the changes, you need to run the upgrade script once again. This procedure is described in the "Multi-language support for URL aliases" section of the 3.10.0 upgrade documentation. However, there are two differences for eZ Publish 4.0:

1. The default value of "TransformationGroup" setting was changed in eZ Publish 4.0 from "urlaliases_iri" to "urlaliases".
2. In eZ Publish 4.0, the script supports some new parameters, and thus you need to use the following shell command to run the script:

```
php bin/php/updateniceurls.php --import --fetch-limit=number -s example
```

Replace "example" with the actual name of the siteaccess and specify the desired number of items to handle per one iteration instead of "number"; the script will automatically do as many iterations as necessary to convert all the URLs, based on the number specified.

The "--import" parameter makes the script go through all entries in the "ezurlaliases" database table, which was used for storing URL aliases in eZ Publish 3.9 and earlier versions. As a result, old URL aliases will be imported to the new system and stored in the "ezurlaliases_ml" table. This database table is used for storing multilingual URL aliases in eZ Publish 3.10.0 and later versions.

If the optional "--fetch-limit" parameter is omitted, the script will handle 200 items per one iteration.

Changes to roles and policies (optional)

If you are running eZ Publish on Windows, note that in eZ Publish 3.10.0 there is a bug in the "updatetipafriendpolicy.php" script that prevents it from granting access to the "Tip a friend" feature to the users. The same bug exists in eZ Publish 3.9.4 and earlier versions (refer to <http://issues.ez.no/11663> for more information). The updated version of the "updatetipafriendpolicy.php" script is available in eZ Publish 4.0.0 (plus the upcoming versions 3.10.1 and 3.9.5). Windows users are encouraged to re-run this script when upgrading from 3.10.0 to 4.0. The following example shows how this can be done:

1. Navigate into the eZ Publish 4.0.0 directory.
2. Run the script using the following shell command:

```
php update/common/scripts/4.0/  
updatetipafriendpolicy.php -s example_admin -l login -p password
```

Replace "example_admin" with the actual name of your admin siteaccess and use the administrator's login and password instead of "login" and "password".

The script will add a new role that grants access to the "Tip a friend" feature and ask for confirmation about assigning this role to each user or user group located one level beneath the "Users" top level node.

Improved handling of country dependent VATs (optional)

The "updatevatcountries.php" upgrade script in eZ Publish 3.10.0 contains a bug. The same bug exists in eZ Publish 3.9.4 and 3.9.3 (refer to <http://issues.ez.no/11955> for more information). The updated version of the script is available for download and will be included in the upcoming eZ Publish 4.0.1, 3.10.1 and 3.9.5 releases. If you upgraded to 3.10.0 using the original version of the script, you need to download the updated version from <http://pubsvn.ez.no> and run it when upgrading from 3.10.0 to 4.0.0.

This script should be run once for each database, specifying one siteaccess per database. If you only have a public and an administration siteaccess that share the same database (which is the most typical/usual case), you only need to run the script for one of the siteaccesses. If the siteaccess isn't specified, the default siteaccess will be used.

The following example shows how to run the script:

1. Navigate into the eZ Publish 4.0.0 directory.
2. Run the "updatevatcountries.php" script (replace "example" with the actual name of your siteaccess):

```
php update/common/scripts/4.0/updatevatcountries.php -s example
```

The script will go through all the VAT rules in the database and update them.

Step 4: Updating the system configuration

Custom/override templates location

In eZ Publish versions prior to 4.0.0, the order of design resources used by the "ezdesign" and "ezimage" template operators was not the same as used by the "include" template function and eZTemplate::fetch(). This might lead to problems with overriding templates in extensions (refer to <http://issues.ez.no/9595> for more information). This issue was fixed in eZ Publish 4.0.0.

In eZ Publish 4.0, templates that override default ones must be placed in the "override/templates" directory of your design. If you are not sure about the location of override templates in your eZ Publish installation, review the settings in your "override.ini.append.php" configuration file. (Refer to "The template override system" for more information.) The names of the override templates are specified using the "MatchFile" directive, which makes it possible to find the corresponding files on a filesystem.

Note that custom templates that are not connected with the override system must be placed in the "templates" directory of a custom design.

Outgoing mail delivery

In eZ Publish 4.0.0 there is a bug in the "ezsmtp" class that prevents the system from sending outgoing mails through an SMTP server. This bug will be fixed in the upcoming eZ Publish 4.0.1 release. If your site uses an SMTP relay server for mail delivery, you need to either switch to sendmail (if available) or [download the updated version of the "ezsmtp" class](#) and replace the corresponding file in your eZ Publish installation.

Step 5: Clearing the caches

Whenever an eZ Publish solution is upgraded, all caches must be cleared in a proper way. This should be done from within a system shell:

1. Navigate into the eZ Publish 4.0 directory.
2. Run the script using the following shell command:

```
php bin/php/ezcache.php --clear-all --purge
```

Purging ensures that the caches are physically removed. Without the "--purge" option, the caches will be expired but not removed.

Sometimes the script is unable to clear all cache files because of restrictive file/directory permission settings. Make sure that all cache files have been cleared by inspecting the contents of the various cache subdirectories within the "var" directory (typically "var/cache/" and "var/<name_of_siteaccess>/cache/" directories). If there are any cache files left, you need to remove them manually.

1.14.5 from 4.0.0 to 4.0.1rc2 (deprecated)

This documentation page is deprecated and no longer maintained. Refer to the following documentation pages for upgrade instructions:

- [Upgrading from 3.10.x to 4.0.y \(page 212\)](#)
- [Upgrading from 4.0.x to 4.0.y \(page 200\)](#)

This section describes how to upgrade your existing eZ Publish 4.0.0 installation to version 4.0.1rc2 (hereinafter called "4.0.1" for the sake of simplicity).

Important note

Before continuing, note that eZ Publish 4.0.1 does not support migrating global URL aliases yet. This means that your global URL aliases (if any) will be lost after upgrading to 4.0.1rc2. You can check the list of global URL aliases in the "Setup - URL translator" part of the administration interface. If there are some items on the list, it is recommended to delay upgrading until the stable release of eZ Publish 4.0.1 is out. (Otherwise, you'll need to re-create all global URL aliases manually after upgrading to 4.0.1rc2.)

Make sure that you have a working backup of the site before you do the actual upgrade. The upgrade procedure consists of the following steps:

1. Upgrading the distribution files to 4.0.1
2. Upgrading the database to 4.0.1
3. Running the system upgrade scripts
4. Updating the system configuration
5. Clearing the caches

Step 1: Upgrading the distribution files

The easiest way to upgrade the distribution files is to unpack eZ Publish 4.0.1 to a separate directory and then copy the directories that contain site-specific files from the existing installation. Make sure that you copy the following directories:

- design/example
- design/example_admin
- var
- settings/siteaccess

- settings/override

Replace "example" and "example_admin" with actual names used by your siteaccesses.

Custom extensions

If you are using custom extensions, the subdirectories inside the "extension" directory will also have to be copied. However, make sure that you do not overwrite any extensions that come with eZ Publish (currently "ezdhtml", "ezodf" and "ezurlaliasmigration"). Note that upgrading the distribution files will overwrite the autoload arrays for extensions. You will need to re-generate the autoload arrays for active extensions later (after enabling the "ezurlaliasmigration" extension in step 3).

If you are using the Website Interface front-end, be sure to copy the "extension/ezwebin/" directory.

Important bug fix for remote IDs

Due to some bugs in old versions of eZ Publish (3.7.3 and others), it might happen that not all content objects in your database have unique remote IDs. This basically means that the "remote_id" column of the "ezcontentobject" table in the database might contain duplicate values. In eZ Publish 4.0.1, a database schema change was introduced in order to ensure that only unique remote IDs can be stored in this table. Because of this, you need to make sure that there are no content objects with identical remote IDs before upgrading the database. This can be done by running the "fixobjectremoteid.php" upgrade script. (Note that both the database schema change and the upgrade script will be added to the upcoming versions 3.9.5 and 3.10.1 as well.)

This script should be run once for each database, specifying one siteaccess per database. If you only have a public and an administration siteaccess that share the same database (which is the most typical/usual case), you only need to run the script for one of the siteaccesses. If the siteaccess isn't specified, the default siteaccess will be used.

The following example shows how to run the script:

1. Navigate into the eZ Publish 4.0.1 directory.
2. Run the script (replace "example" with the actual name of your siteaccess):

```
php update/common/scripts/4.0/fixobjectremoteid.php -s example
```

The script will search for objects with non-unique remote IDs. Every time such an object is found, the script will suggest to either display more detailed information or fix the problem automatically. If you choose the first option, the script will display the list of objects that have the same remote ID and let you decide which one should remain unchanged; all other objects in the list will get new remote IDs. Otherwise, the script will fix the problem automatically based

on the objects' creation dates. This means that the script will generate new remote IDs for all objects in the list except the one that was created first.

The optional "--mode" parameter makes it possible to run the script in either "auto-fix" or "manual-fix" mode as shown below.

```
php update/common/scripts/4.0/fixobjectremoteid.php -s example --mode=value
```

If you replace "value" with "a" in the command above, the script will automatically fix all non-unique remote IDs that are found in the database based on the objects' creation dates. In this case, no further input is required from the user. If you specify "--mode=d", the script will display the list of objects for each non-unique remote ID so that you can manually choose which objects should keep their remote IDs.

Step 2: Upgrading the database

The following text describes how a 4.0.0 database can be upgraded to 4.0.1. (Note that even though the sql file contains a section marked with "from 3.10.1" comments, you do not need to skip this part when upgrading from 4.0.0 to 4.0.1.)

MySQL

1. Navigate into the eZ Publish 4.0.1 directory.
2. Run the database upgrade script:

```
mysql -u <username> -p<password> <database> < update/database/mysql/4.0/  
dbupdate-4.0.0-to-4.0.1.sql
```

PostgreSQL

1. Navigate into the eZ Publish 4.0.1 directory.
2. Run the database upgrade script:

```
psql -d <database> -U <downer> < update/database/postgresql/4.0/  
dbupdate-4.0.0-to-4.0.1.sql
```

Step 3: Running the system upgrade scripts

The 4.0.1 version of eZ Publish introduces a couple of important bug fixes and functionality changes. In order to make sure that your site is compatible with these changes, you may need to run a few upgrade scripts.

Multi-language support for URL aliases

In eZ Publish 4.0.1, the multi-language support for URL aliases was substantially improved, along with some important bug fixes related to this functionality. When upgrading, you need to regenerate all URL aliases that are stored in the "ezurlalias_ml" database table. This basically means that all auto-generated URL aliases will be deleted and then created from scratch by using the "updateniceurls.php" script. However, manual/user-defined URL aliases and URL history entries cannot be regenerated automatically. Instead, these items will be transferred/migrated into a temporary table in the database (hereinafter called "migration table") and then restored at a later stage. This must be done by using the "migrate.php" script, which is included in the "ezurlalias-migration" extension. This new extension is included in eZ Publish 4.0.1 and provides tools for the purpose of URL alias migration.

Follow the instructions below in order to update your URL aliases.

1. Create a database backup.

Before you continue, create a backup of your database (or at least make sure that you have a backup of the "ezurlalias_ml" table itself).

2. Enable the "ezurlaliasmigration" extension.

The "ezurlaliasmigration" extension must be enabled for all of your siteaccesses. To do this, edit the "site.ini.append.php" file located in the "settings/override" directory and add the following line under the [ExtensionSettings] section:

```
ActiveExtensions []=ezurlaliasmigration
```

Alternatively, you can enable the extension separately for each siteaccess. To do this, edit the "site.ini.append.php" file located in the "settings/siteaccess/example" directory (replace "example" with the name of the target siteaccess) and add the following line under the [ExtensionSettings] section:

```
ActiveAccessExtensions []=ezurlaliasmigration
```

3. Update the autoload arrays for extensions (optional).

If you are using custom extensions, some of them might stop working after upgrading to 4.0.1. The reason for this is that the "autoload/ezp_extension.php" file is overwritten in the first step of the upgrade procedure (when upgrading the distribution files). To solve the problem, you need to regenerate the autoload arrays for active extensions as described below.

Note that the default "autoload/ezp_extension.php" file in eZ Publish 4.0.1 contains all the information needed for the following extensions to work correctly: "ezdhtml", "ezodf", "ezurlaliasmigration", "ezflow" and "ezwebin". If you are only using these extensions (no custom ones), you do not have to regenerate the autoload arrays when upgrading to 4.0.1.

The following example shows how to regenerate the autoload arrays for active extensions.

1. Navigate into the eZ Publish 4.0.1 directory.
2. Run the script using the following shell command:

```
php bin/php/ezpgenerateautoloads.php --extension
```

The script will look for class definitions in the "extension" directory and update the "autoload/ezp_extension.php" file accordingly.

4. Create the migration table.

The "ezurlaliasmigration" extension requires a custom database table called "ezurlalias_ml_migrate" to be created in the database. This table will be used for storing the migrated URL alias data. The new table must be created in accordance with the definitions specified in the "schema.sql" file included in the extension. This can be done by executing the SQL queries from the "schema.sql" file as described below.

If you are using MySQL, navigate into the eZ Publish 4.0.1 directory and run the following shell command:

```
mysql -u <username> -p<password> <database> < extension/ezurlaliasmigration/  
sql/mysql/schema.sql
```

If you are running PostgreSQL, use the following shell command instead:

```
psql -d <database> -U <dbowner> < extension/ezurlaliasmigration/sql/postgresql/  
schema.sql
```

Alternatively, you can use the "migrate.php" script to create the migration table. This option will work regardless of which database implementation you are using. You need to navigate into the eZ Publish 4.0.1 directory and run the script from within the system shell:

```
php extension/ezurlaliasmigration/scripts/migrate.php --create-migration-table
```

With this command, the script will create the "ezurlalias_ml_migrate" table in the database that is used by the default siteaccess. If your site makes use of two or more databases, you need to run the script once for each database, specifying one siteaccess per database. Use the "-s" parameter to specify the target siteaccess as shown below:

```
php extension/ezurlaliasmigration/scripts/  
migrate.php -s example --create-migration-table
```

(Replace "example" with the actual name of your siteaccess.)

5. Transfer your URL aliases to the migration table.

To find out about the current number of manual/user-defined URL aliases and URL history entries in your system, run the "migrate.php" script without parameters or with the "-s" parameter only. The following example shows how this can be done.

1. Navigate into the eZ Publish 4.0.1 directory.
2. Run the script (replace "example" with the actual name of your siteaccess):

```
php extension/ezurlaliasmigration/scripts/migrate.php -s example
```

Note that the commands for running the "migrate.php" and "updateniceurls.php" scripts in the examples below will assume that all your siteaccesses share the same database. Because of this, the "-s example" part will be omitted and the scripts will be run for the database that is used by the default siteaccess. If you are using two or more databases, the "migrate.php" script should be run once for each database, specifying one siteaccess per database (by adding the "-s" parameter to each command). This also applies to the "updateniceurls.php" script. Both scripts need to be run from the root directory of your eZ Publish 4.0.1 installation. The following table reveals the available parameters for the "migrate.php" script.

Parameter	Description
--create-migration-table	Create the migration table in the database.
--migrate-alias	Transfer manual URL aliases to the migration table.
--migrate-history	Transfer URL history entries to the migration table.
--migrate	Transfer both manual URL aliases and URL history entries to the migration table.
--restore-alias	Restore manual URL aliases from the migration table.
--restore-history	Restore URL history entries from the migration table.
--restore	Restore both manual URL aliases and URL history entries from the migration table.

To transfer your manual URL aliases and URL history entries into the migration table, run the "migrate.php" script as shown below:

```
php extension/ezurlaliasmigration/scripts/migrate.php --migrate
```

The script will go through all manual URL aliases and URL history entries in the "ezurlalias_ml" table and put the collected data into the "ezurlalias_ml_migrate" table.

Instead of transferring both manual URL aliases and URL history entries at once, it is also possible to run the script once for each type of URL entries. To do this, use either the "--migrate-alias" or

"--migrate-history" parameter instead of "--migrate" when running the script (refer to the table above for more information about these parameters).

It is important not to change any configuration settings after transferring the data to the migration table. The configuration of the URL alias system must remain untouched until you're finished with regenerating the URL aliases and transferring the data back to the "ezurlalias_ml" table (otherwise, there might be some broken or inconsistent URLs on your site after upgrading to 4.0.1).

6. Remove old URL alias data.

Use the following SQL query to empty the "ezurlalias_ml" table in your database:

```
TRUNCATE ezurlalias_ml;
```

7. Create new auto-generated URL aliases.

Now that the "ezurlalias_ml" table is empty, you need to create new auto-generated URL aliases based on the existing node tree structure and content object names. To do this, run the "updateniceurls.php" script using the following command:

```
php bin/php/updateniceurls.php --fetch-limit=number
```

Specify the desired number of items to handle per one iteration instead of "number". On a system with a 128 MB memory limit for PHP, this parameter can be set to 2000. (The script will automatically do as many iterations as necessary to generate the URL aliases for all nodes, based on the number specified. If the optional "--fetch-limit" parameter is omitted, the script will handle 200 items per one iteration.)

The script will go through all content objects stored in the database and create new virtual URLs for them in accordance with the current transformation settings. New URL aliases will be generated for all of the translation languages and placed in the "ezurlalias_ml" table in the database.

8. Restore migrated URL data.

You need to restore the custom URL aliases and URL history entries from the migration table. This can be done by running the "migrate.php" script:

```
php extension/ezurlaliasmigration/scripts/migrate.php --restore
```

The script will go through all entries in the "ezurlalias_ml_migrate" table and put the collected data back into the "ezurlalias_ml" table.

Instead of restoring both manual URL aliases and URL history entries at once, it is also possible to run the script once for each type of URL entries. To do this, use either the "--restore-alias"

or "--restore-history" parameter instead of "--restore" when running the script (refer to the table above for more information about these parameters).

It might happen that a URL alias entry was previously corrupted and cannot be automatically restored from the migration table. In this case, you will see "F" in the console output of the "migrate.php" script. The following SQL query can be used in order to display the migrated entries that were not restored:

```
SELECT id, link, parent, lang_mask, text, action, is_original, is_alias,
lang_mask_adjusted FROM ezurlalias_ml_migrate WHERE is_restored=0;
```

If you want the URL entries that were not restored automatically to work on your site, you must create new user-defined URLs later (when the upgrade procedure is finished). This can be done from within the administration interface as described on the "Managing URL aliases" documentation page.

Note that the "migrate.php" script is currently unable to restore global URL aliases from the migration table (refer to <http://issues.ez.no/13475> for more information). This issue will be fixed in the stable release of version 4.0.1.

Changes to system upgrade scripts (optional)

If your solution was initially built on eZ Publish 3.10 or 4.0, skip this part. Otherwise, read it carefully.

The "updatevatcountries.php" upgrade script in eZ Publish 3.10.0 contains a bug. The same bug exists in eZ Publish 3.9.3 and 3.9.4 (refer to <http://issues.ez.no/11955> for more information). The updated version of the script is available in eZ Publish 4.0.1 (plus the upcoming versions 3.9.5 and 3.10.1). If you previously upgraded to 3.10.0 using the original version of the script and then didn't run the updated version when upgrading from 3.10.0 to 4.0.0, you need to run it when upgrading to 4.0.1.

This script should be run once for each database, specifying one siteaccess per database. If you only have a public and an administration siteaccess that share the same database (which is the most typical/usual case), you only need to run the script for one of the siteaccesses. If the siteaccess isn't specified, the default siteaccess will be used.

The following example shows how to run the script:

1. Navigate into the eZ Publish 4.0.1 directory.
2. Run the script (replace "example" with the actual name of your siteaccess):

```
php update/common/scripts/4.0/updatevatcountries.php -s example
```

The script will go through all the VAT rules in the database and update them.

Step 4: Updating the system configuration

You are not required to do any configuration changes when upgrading from 4.0.0 to 4.0.1.

Step 5: Clearing the caches

Whenever an eZ Publish solution is upgraded, all caches must be cleared in a proper way. This should be done from within a system shell:

1. Navigate into the eZ Publish 4.0.1 directory.
2. Run the script using the following shell command:

```
php bin/php/ezcache.php --clear-all --purge
```

Purging ensures that the caches are physically removed. When the "--purge" parameter is not specified, the caches will be expired but not removed (refer to this page for more information).

Sometimes the script is unable to clear all cache files because of restrictive file/directory permission settings. Make sure that all cache files have been cleared by inspecting the contents of the various cache subdirectories within the "var" directory (typically the "var/cache/" and "var/<name_of_siteaccess>/cache/" directories). If there are any cache files left, you need to remove them manually.

1.15 Upgrading to 3.10

This chapter contains information related to the upgrade of an eZ Publish solution to [version 3.10](#). Follow the upgrade procedures that are described in the following sections:

- [Upgrading from 3.10.x to 3.10.y \(page 235\)](#)
- [Upgrading from 3.9.x to 3.10.y \(page 242\)](#)

1.15.1 from 3.10.x to 3.10.y

This section describes how to upgrade your existing eZ Publish 3.10.x installation to version 3.10.y, for example from 3.10.0 to 3.10.1. If you are upgrading from a version prior to eZ Publish 3.10.0, refer to the "Upgrading from 3.9.x to 3.10.y (page 242)" section for upgrade instructions.

It is recommended to upgrade directly to the latest stable release in the 3.10 branch, which contains all important bug fixes. Refer to the [changelogs](#) and [security advisories](#) for more information about the issues that were fixed in the latest releases or view the short list of changes here.

The upgrade procedure described below is generic and does not cover any specific cases. If you are running eZ Publish in a clustered environment, refer to the [developer documentation](#) for upgrade instructions. In case the eZ Publish Extension for Oracle Database is used, refer to the documentation of the database extension.

Make sure that you have a working backup of the site before you do the actual upgrade. The upgrade procedure consists of the following steps:

1. Upgrading the distribution files to 3.10.1
2. Upgrading the database to 3.10.1
3. Running the system upgrade scripts
4. Updating the system configuration
5. Clearing the caches

Step 1: Upgrading the distribution files

The easiest way to upgrade the distribution files is to unpack eZ Publish 3.10.1 to a directory and then copy the directories that contain site-specific files from the existing installation. Make sure that you copy the following directories:

- design/example
- design/example_admin
- var
- settings/siteaccess
- settings/override

Replace "example" and "example_admin" with the actual names of your siteaccesses.

Custom extensions

If you are using custom extensions, the subdirectories inside the "extension" directory will also have to be copied. However, make sure that you do not overwrite any extensions that come with eZ Publish (currently "ezdhtml", "ezodf", "ezpaypal" and "ezurlaliasmigration").

If you are using the Website Interface front-end, be sure to copy the "extension/ezwebin/" directory. It is also possible to upgrade the Website Interface extension to a newer version (you can find more information and instructions in the "Upgrading the Website Interface" chapter of the Website Interface installation guide).

Important bug fix for remote IDs

Due to some bugs in previous versions of eZ Publish (3.9.0 and others), it might happen that not all content objects in your database have unique remote IDs. This basically means that the "remote_id" column of the "ezcontentobject" table in the database might contain duplicate values. In eZ Publish 3.10.1, a database schema change was introduced in order to ensure that only unique remote IDs can be stored in this table. Because of this, you need to make sure that there are no content objects with identical remote IDs before upgrading the database. This can be done by running the "fixobjectremoteid.php" upgrade script. (Note that both the database schema change and the upgrade script were added to eZ Publish 3.9.5 and 4.0.1 as well.)

This script should be run once for each database, specifying one siteaccess per database. If you only have a public and an administration siteaccess that share the same database (which is the most typical/usual case), you only need to run the script for one of the siteaccesses. If the siteaccess isn't specified, the default siteaccess will be used.

The following example shows how to run the script:

1. Navigate into the eZ Publish 3.10.1 directory.
2. Run the script (replace "example" with the actual name of your siteaccess):

```
php update/common/scripts/3.10/fixobjectremoteid.php -s example
```

The script will search for objects with non-unique remote IDs. Every time such an object is found, the script will suggest to either display more detailed information or fix the problem automatically. If you choose the first option, the script will display the list of objects that have the same remote ID and let you decide which one should remain unchanged; all other objects in the list will get new remote IDs. Otherwise, the script will fix the problem automatically based on the objects' creation dates. This means that the script will generate new remote IDs for all objects in the list except the one that was created first.

The optional "--mode" parameter makes it possible to run the script in either "auto-fix" or "manual-fix" mode as shown below.

```
php update/common/scripts/3.10/fixobjectremoteid.php -s example --mode=value
```

If you replace "value" with "a" in the command above, the script will automatically fix all non-unique remote IDs that are found in the database based on the objects' creation dates. In this case, no further input is required from the user. If you specify "--mode=d", the script will display the list of objects for each non-unique remote ID so that you can manually choose which objects should keep their remote IDs.

Step 2: Upgrading the database

The following text describes how a 3.10.0 database can be upgraded to 3.10.1.

MySQL

1. Navigate into the eZ Publish 3.10.1 directory.
2. Run the database upgrade script:

```
mysql -u <username> -p<password> <database> < update/database/mysql/3.10/  
dbupdate-3.10.0-to-3.10.1.sql
```

Note that the CREATE TABLE statement in the database upgrade script does not specify which storage engine to use (no ENGINE or TYPE option), and thus the default storage engine will be used. Normally, it is MyISAM (starting from MySQL v.3.23). If you are using InnoDB, make sure the default storage engine is set to InnoDB before you run the database upgrade script (refer to the [MySQL documentation](#) for information about how to set the default engine). If you were not able to change the MySQL configuration on your server, and the upgrade left you with a mix of table types, you can convert the newly created table to InnoDB using the following SQL query:

```
ALTER TABLE ezurlwildcard TYPE = innodb;
```

It is also possible to use the "bin/php/ezconvertmysqltabletype.php" script for database conversion.

PostgreSQL

1. Navigate into the eZ Publish 3.10.1 directory.
2. Run the database upgrade script:

```
psql -d <database> -U <dbowner> < update/database/postgresql/3.10/  
dbupdate-3.10.0-to-3.10.1.sql
```

Step 3: Running the system upgrade scripts

The 3.10.1 version of eZ Publish introduces a couple of important bug fixes and functionality changes. In order to make sure that your site is compatible with these changes, you may need to run a few upgrade scripts.

Multi-language support for URL aliases

In eZ Publish 3.10.1, the multi-language support for URL aliases was substantially improved, along with some important bug fixes related to this functionality. In addition, the "bin/php/updateniceurls.php" script was updated and the support for new parameters was added. When upgrading, you need to regenerate all URL aliases that are stored in the "ezurlalias_ml" database table. This basically means that all auto-generated URL aliases will be deleted and then created from scratch by using the "updateniceurls.php" script. However, manual/user-defined URL aliases and URL history entries cannot be regenerated automatically. Instead, these items will be transferred/migrated into a temporary table in the database (hereinafter called "migration table") and then restored at a later stage. This must be done by using the "migrate.php" script, which is included in the "ezurlaliasmigration" extension. This new extension is included in eZ Publish 3.10.1 and provides tools for the purpose of URL alias migration. (Note that the "ezurlaliasmigration" extension was added to eZ Publish 4.0.1 as well.)

Follow the instructions below in order to update your URL aliases.

1. Create a database backup.

Before you continue, create a backup of your database (or at least make sure that you have a backup of the "ezurlalias_ml" table itself).

2. Review your configuration settings.

In eZ Publish 3.10.1 the default type of URL transformation was changed from "urlalias_iri" to "urlalias". Because of this, you must review your URL alias configuration settings as described below.

- If the "TransformationGroup" directive is present in either global override for "site.ini" or "settings/siteaccess/<your_siteaccess>/site.ini.append.php" for all siteaccesses, no configuration changes are required.
- Otherwise, your site makes use of the default type of URL transformation. In eZ Publish 3.10.0, it was "urlalias_iri". When upgrading to 3.10.1, you must explicitly specify "urlalias_iri" using the "TransformationGroup" directive located in the [URLTranslator] section of either "settings/siteaccess/<your_siteaccess>/site.ini.append.php" (for each siteaccess) or "settings/override/site.ini.append.php" (if all your siteaccesses are using the same database).

In other words, make sure that your existing URL alias configuration does not change because of the new default value of the "TransformationGroup" directive introduced in 3.10.1. Without this, there might be some broken or inconsistent URLs on your site after upgrading to 3.10.1.

3. Enable the "ezurlaliasmigration" extension.

The "ezurlaliasmigration" extension must be enabled for all of your siteaccesses. To do this, edit the "site.ini.append.php" file located in the "settings/override" directory and add the following line under the [ExtensionSettings] section:

```
ActiveExtensions []=ezurlaliasmigration
```

Alternatively, you can enable the extension separately for each siteaccess. To do this, edit the "site.ini.append.php" file located in the "settings/siteaccess/example" directory (replace "example" with the name of the target siteaccess) and add the following line under the [ExtensionSettings] section:

```
ActiveAccessExtensions []=ezurlaliasmigration
```

4. Proceed with updating your URL aliases.

In order to update your URL aliases, you need to do the following:

- Create the migration table.
- Transfer your URL aliases to the migration table.
- Remove old URL alias data.
- Create new auto-generated URL aliases.
- Restore migrated URL data.

When upgrading from 3.10.0 to 3.10.1, these operations are exactly the same as when upgrading from version 4.0.0 to 4.0.1. Read the corresponding subsections of the "Upgrading from 4.0.x to 4.0.y" documentation page starting from "4. Create the migration table".

Binary files uploaded via WebDAV (optional)

In eZ Publish 3.10.0, there is a bug in the "File" datatype which may cause problems with binary file uploading via WebDAV. The typical symptoms are that the file extension of the uploaded file is missing and that you may not be able to download the file once it is uploaded (refer to <http://issues.ez.no/9450> for more information). The same bug exists in eZ Publish 3.9.4 and earlier versions. This issue was fixed in eZ Publish 3.9.5, 3.10.1 and 4.0.0.

If your site has objects that were created by uploading files via WebDAV and you have configured MIME-type-to-class mapping using the "MimeClassMap[]" directive in an override for "upload.ini", you need to run the "updatebinaryfile.php" script in order to fix the file extensions. (If a content class does not have any attributes that make use of the "File" datatype, instances of this class are not affected.)

Refer to the corresponding part of the "Upgrading from 3.9.x to 3.10.y" documentation page for instructions about how to run the "updatebinaryfile.php" script.

Changes to system upgrade scripts (optional)

If your solution was initially built on eZ Publish 3.9 or an earlier version and then upgraded to 3.10.0, read this part carefully. Otherwise, skip this part.

The "updatevatcountries.php" upgrade script in eZ Publish 3.10.0 contains a bug. The same bug exists in eZ Publish 3.9.4 and 3.9.3 (refer to <http://issues.ez.no/11955> for more information). The updated version of the script is available in eZ Publish 3.9.5, 3.10.1 and 4.0.0. If you previously upgraded to 3.10.0 using the original version of the script, you need to run the updated version of the script when upgrading from 3.10.0 to 3.10.1. Refer to the corresponding part of the "Upgrading from 3.9.x to 3.10.y" documentation page for instructions about how to run the "updatevatcountries.php" script.

If you are running eZ Publish on Windows, note that in eZ Publish 3.10.0 there is a bug in the "updatetipafriendpolicy.php" upgrade script that prevents it from granting access to the "Tip a friend" feature to the users. The same bug exists in eZ Publish 3.9.3 - 3.9.4 and 3.8.9 - 3.8.10 (refer to <http://issues.ez.no/11663> for more information). The updated version of the script is available in eZ Publish 3.9.5, 3.10.1 and 4.0.0. Windows users that previously upgraded to 3.10.0 using the original version of the script are encouraged to re-run this script when upgrading from 3.10.0 to 3.10.1. Refer to the corresponding part of the "Upgrading from 3.9.x to 3.10.y" documentation page for instructions about how to run the "updatetipafriendpolicy.php" script.

Step 4: Updating the system configuration

You are not required to do any configuration changes when upgrading from 3.10.0 to 3.10.1.

Tree menu configuration (optional)

eZ Publish 3.10.1 introduces some changes to the dynamic tree menu functionality. In particular, the "treemenu" view of the "content" module no longer makes use of parameters that are transferred using GET variables (this improves tree menu compatibility when running PHP in [FastCGI](#) mode; refer to <http://issues.ez.no/11806> for more information).

If you have instructed Apache to use "index_treemenu.php" instead of "index.php" when the treemenu view of the content module is requested, this feature will stop working after upgrading to version 3.10.1. To make it work, update your rewrite rules as described below.

If your ".htaccess" file contains the following line:

```
RewriteRule content/treemenu/?$ index_treemenu.php
```

you need to remove "\$" from there. The updated line should look like this:

```
RewriteRule content/treemenu/? index_treemenu.php
```

If no ".htaccess" file is used, check whether the following line is present in your Apache configuration file:

```
RewriteRule content/treemenu/?$ /index_treemenu.php [L]
```

If found, remove "\$" from this line. The result will look like this:

```
RewriteRule content/treemenu/? /index_treemenu.php [L]
```

Note that due to a bug in eZ Publish 3.10.1, you need to download the updated version of "index_treemenu.php" from <http://pubsvn.ez.no> and replace the corresponding file in your installation. The same [bug](#) exists in eZ Publish 4.0.1.

Step 5: Clearing the caches

Whenever an eZ Publish solution is upgraded, all caches must be cleared in a proper way. This should be done from within a system shell:

1. Navigate into the eZ Publish 3.10.1 directory.
2. Run the script using the following shell command:

```
php bin/php/ezcache.php --clear-all --purge
```

Purging ensures that the caches are physically removed. When the "--purge" parameter is not specified, the caches will be expired but not removed.

Sometimes the script is unable to clear all cache files because of restrictive file/directory permission settings. Make sure that all cache files have been cleared by inspecting the contents of the various cache subdirectories within the "var" directory (typically "var/cache/" and "var/<name_of_siteaccess>/cache/" directories). If there are any cache files left, you need to remove them manually.

1.15.2 from 3.9.x to 3.10.y

This section describes how to upgrade your existing eZ Publish 3.9.x installation to version 3.10.y (but not 3.10.0, since direct upgrade to this version is not recommended). If you are upgrading from a version prior to eZ Publish 3.9, you need to first upgrade to 3.9 (page 264) before you can upgrade to 3.10.

The upgrade procedure described below is generic and does not cover any specific cases. If you are running eZ Publish in a clustered environment, refer to the developer documentation for upgrade instructions. In case the eZ Publish Extension for Oracle Database is used, refer to the documentation of the database extension.

Important note

Before continuing, note that it is strongly recommended to upgrade directly to the latest stable release in the 3.10 branch (3.10.1 at the time of writing), which contains all important bug fixes. Refer to the changelogs and [security advisories](#) for more information about the issues that were fixed in the latest releases or view the short list of changes below.

- In eZ Publish 3.10.1, the multi-language support for URL aliases was substantially improved, along with some important bug fixes related to this functionality. In addition, the support for wildcard based URL forwarding was re-added and the default type of URL transformation was changed to "urlalias". Refer to <http://issues.ez.no/12785>, <http://issues.ez.no/11650> and <http://issues.ez.no/12093> for more information.
- The "updateipafriendpolicy.php" upgrade script in eZ Publish 3.10.0 contains a bug. The same bug exists in eZ Publish 3.9.4 and 3.9.3. It prevents the script from granting access to the "Tip a friend" feature to users. The problem occurs when the script is run on Windows. This issue was fixed in eZ Publish 3.9.5, 3.10.1 and 4.0.0. Refer to <http://issues.ez.no/11663> for more information.
- The "updatevatcountries.php" upgrade script in eZ Publish 3.10.0 contains a bug. The same bug exists in eZ Publish 3.9.3 and 3.9.4. This issue was fixed in eZ Publish 3.9.5, 3.10.1 and 4.0.1. Refer to <http://issues.ez.no/11955> for more information.
- In eZ Publish 3.10.0, there is a bug in the "File" datatype which may cause problems with binary file uploading via WebDAV. The same bug exists in eZ Publish 3.9.0 - 3.9.4. This issue was fixed in eZ Publish 3.9.5, 3.10.1 and 4.0.0. Refer to <http://issues.ez.no/9450> for more information.
- In eZ Publish 3.10.0, the "register" view of the "user" module contains a bug. It prevents the system from registering new users when the "VerifyUserEmail" setting is enabled (it is enabled by default). The same bug exists in eZ Publish 3.8.10 and 3.9.4. This issue was fixed in eZ Publish 3.9.5 and 3.10.1. Refer to <http://issues.ez.no/11598> for more information.

Make sure that you have a working backup of the site before you do the actual upgrade. The upgrade procedure consists of the following steps:

1. Upgrading the distribution files to 3.10
2. Upgrading the database to 3.10
3. Running the 3.10 upgrade scripts
4. Updating the system configuration
5. Clearing the caches

Step 1: Upgrading the distribution files

The easiest way to upgrade the distribution files is to unpack eZ Publish 3.10.1 to a directory and then copy the directories that contain site-specific files from the existing installation. Make sure that you copy the following directories:

- design/example
- design/example_admin
- var
- settings/siteaccess
- settings/override

Replace "example" and "example_admin" with the actual names of your siteaccesses.

Custom extensions

If you are using custom extensions, the subdirectories inside the "extension" directory will also have to be copied. However, make sure that you do not overwrite any extensions that come with eZ Publish (currently "ezdhtml", "ezodf", "ezpaypal" and "ezurlaliasmigration").

If you are using the Website Interface front-end, make sure you copied the "extension/ezwebin/" directory. It is also possible to upgrade the Website Interface extension to a newer version (you can find more information and instructions in the "Upgrading the Website Interface" chapter of the Website Interface installation guide).

Translations

eZ Publish 3.10 has less translations than 3.9 releases because not all of the translation contributors have submitted a [CLA](#) yet. If you are using a multilingual site the possible workarounds in that case can be either to use the 3.9.x translations or to download translations from <http://ez.no/developer/translations>. It would also be highly appreciated if translators for existing translations send us the CLA so that we can include the translations in future 3.10 releases.

Important bug fix for remote IDs

Due to some bugs in previous versions of eZ Publish (3.9.0 and others), it might happen that not all content objects in your database have unique remote IDs. This basically means that the "remote_id" column of the "ezcontentobject" table in the database might contain duplicate values. In eZ Publish 3.10.1, a database schema change was introduced in order to ensure that only unique remote IDs can be stored in this table. Because of this, you need to make sure that there are no content objects with identical remote IDs before upgrading the database. This can be done by running the "fixobjectremoteid.php" upgrade script. (Note that both the database schema change and the upgrade script were added to eZ Publish 3.9.5 and 4.0.1 as well.)

This script should be run once for each database, specifying one siteaccess per database. If you only have a public and an administration siteaccess that share the same database (which is the most typical/usual case), you only need to run the script for one of the siteaccesses. If the siteaccess isn't specified, the default siteaccess will be used.

The following example shows how to run the script:

1. Navigate into the eZ Publish 3.10.1 directory.
2. Run the script (replace "example" with the actual name of your siteaccess):

```
php update/common/scripts/3.10/fixobjectremoteid.php -s example
```

The script will search for objects with non-unique remote IDs. Every time such an object is found, the script will suggest to either display more detailed information or fix the problem automatically. If you choose the first option, the script will display the list of objects that have the same remote ID and let you decide which one should remain unchanged; all other objects in the list will get new remote IDs. Otherwise, the script will fix the problem automatically based on the objects' creation dates. This means that the script will generate new remote IDs for all objects in the list except the one that was created first.

The optional "--mode" parameter makes it possible to run the script in either "auto-fix" or "manual-fix" mode as shown below.

```
php update/common/scripts/3.10/fixobjectremoteid.php -s example --mode=value
```

If you replace "value" with "a" in the command above, the script will automatically fix all non-unique remote IDs that are found in the database based on the objects' creation dates. In this case, no further input is required from the user. If you specify "--mode=d", the script will display the list of objects for each non-unique remote ID so that you can manually choose which objects should keep their remote IDs.

Step 2: Upgrading the database

To upgrade a 3.9.x database to 3.10.1, navigate into the eZ Publish 3.10.1 directory and run the following database upgrade scripts one after another:

Multi-language support for URL aliases

In eZ Publish 3.10.0, a new feature has been added that makes it possible to use multi-lingual virtual URLs. In eZ Publish 3.10.1, the multi-language support for URL aliases was substantially improved, along with some important bug fixes related to this functionality. In addition, the support for wildcard based URL forwarding was re-added and the default type of URL transformation was changed to "urlalias" in 3.10.1.

When upgrading from 3.9.x to 3.10.1, you need to specify the desired type of URL transformation using the "TransformationGroup" setting located in the [URLTranslator] section of an override for "site.ini" and run the "updateniceurls.php" script as described below. It is also possible to choose which word separator should be used in the URLs. This can be done by setting the "WordSeparator" configuration directive also located in the [URLTranslator] section to either "dash", "underscore" or "space". Note that this setting will be ignored when the "urlalias_compat" transformation is used (it only supports underscores as separators).

Before running the script, make sure that you have specified the same transformation settings for all siteaccesses that are using the same database. If all your siteaccesses are using the same database (which is the most typical/usual case), you can set the "TransformationGroup" and "WordSeparator" directives in "settings/override/site.ini.append.php". Otherwise, you will have to configure URL transformation from within "settings/siteaccess/example/site.ini.append.php" (where "example" should be replaced by the actual name of your siteaccess) for each siteaccess.

In addition, you must clear the INI cache before running the script, so that the system will re-read the changed configuration files.

The "updateniceurls.php" script should be run once for each database, specifying one siteaccess per database. If you only have a public and an administration siteaccess that share the same database, you only need to run the script for one of the siteaccesses. If the siteaccess is not specified, the default siteaccess will be used.

The following example shows how to run the script.

1. Navigate into the eZ Publish 3.10.1 directory.
2. Run the script using the following shell command:

```
php bin/php/updateniceurls.php --import --fetch-limit=number -s example
```

Replace "example" with the actual name of the siteaccess and specify the desired number of items to handle per one iteration instead of "number". (The script will automatically do as many iterations as necessary to convert all the URLs, based on the number specified. If the optional "--fetch-limit" parameter is omitted, the script will handle 200 items per one iteration.)

The script will go through all content objects stored in the database and create new virtual URLs for them in accordance with the specified transformation settings. Old virtual URLs will be transferred to the new system and stored as history entries. (The "--import" parameter makes the script go through all entries in the "ezurlalias" database table, which was used for storing URL aliases in eZ Publish 3.9. As a result, old URL aliases will be imported to the "ezurlalias_ml" table, which is used for storing multilingual URL aliases in eZ Publish 3.10.)

Note that new URL aliases will be generated for all the languages of your site regardless if they are available or not for the specified siteaccess. The time it takes to update all URL aliases is proportional to the number of nodes on your site.

ISBN datatype enhancements

In eZ Publish 3.10, the handling of ISBN-13 numbers has been significantly improved (refer to the ISBN datatype documentation for details). The different ranges are now stored in the database, allowing the system to fix up incorrectly entered ISBN-13 codes (codes with misplaced or missing dashes). When a new version of eZ Publish is released, the latest information about the different ranges is downloaded from <http://www.isbn-international.org/converter/ranges.js> and stored in a ".dba" file. During the installation process, the setup wizard imports this information to the database. When upgrading, you need to import ranges to the database manually. This must be done even though your site does not use ISBN-13 numbers. You can either import ranges from the ".dba" file using the "ezimportdbafile.php" script or download the latest "range.js" file and run the "updateisbn13.php" script.

The following example shows how to import ranges from the ".dba" file:

1. Navigate into the eZ Publish 3.10.1 directory.
2. Run the script (replace "example" with the actual name of your siteaccess):

```
php bin/php/ezimportdbafile.php --datatype=ezisbn -s example
```

The "datatype" parameter tells the script which datatype the data should be imported for.

The next example shows how to get the latest ranges from the ISBN International website:

1. Download a ".js" file containing the list of ranges using the following link:
<http://www.isbn-international.org/converter/ranges.js>
2. Navigate into the eZ Publish 3.10.1 directory and copy the downloaded file there.
3. Run the "updateisbn13.php" script using the following shell command:

```
php bin/php/updateisbn13.php --file=ranges.js
```

The script will import information from the "range.js" file to the database.

In order to use ISBN-13 numbers on your site after upgrading, you need to do the following:

1. Make sure that the "ISBN-13 format" checkbox in the class attribute edit interface for the corresponding attribute is enabled (it is off by default).
2. Navigate into the eZ Publish 3.10.1 directory.

3. Run the "ezconvert2isbn13.php" script (replace "example" with the actual name of your siteaccess):

```
php bin/php/ezconvert2isbn13.php -s example
```

The script will go through all content object attributes using the "ISBN" datatype in the database and convert their values into valid ISBN-13 numbers. The following table reveals the available options for this script.

Option	Description
--class-id=value	The script will only update attributes of content objects that belong to the specified class. You need to replace "value" with the actual ID number of the class.
--attribute-id=value	The script will only update content object attributes that correspond to the specified content class attribute. You need to replace "value" with the actual ID number of the class attribute.
--all-classes	The script will go through all content objects.
--force	The script will update all attributes of the "ISBN" datatype regardless if the "ISBN-13 format" checkbox is enabled for the class attribute. It will also enable this setting at the class attribute level where it was previously disabled.

These scripts should be run once for each database, specifying one siteaccess per database. If you only have a public and an administration siteaccess that share the same database, you only need to run the scripts for one of the siteaccesses. If the siteaccess is not specified, the default siteaccess will be used.

Enhanced Multi-option datatype

In eZ Publish 3.10, a new datatype called "Multi-option2" has been added in order to replace the old "Multi-option" datatype. The new datatype allows option selections and makes it possible to create multiple and distinctive groups of multi-options for each content object. In addition, it supports rules and provides a handful of new and interesting features. Note that the old "Multi-option" datatype is deprecated and should no longer be used (it will be removed in the future).

In order to be compatible with the new datatype, you will have to run the "updatemultioption.php" script. This script should be run once for each database, specifying one siteaccess per database. If you only have a public and an administration siteaccess that share the same database, you only need to run the script for one of the siteaccesses. If the siteaccess is not specified, the default siteaccess will be used.

The following example shows how to run the script:

1. Navigate into the eZ Publish 3.10.1 directory.
2. Run the script (replace "example" with the actual name of your siteaccess):

```
php update/common/scripts/3.10/updatesmultioption.php -s example
```

The script will go through all content object attributes that are using the old "Multi-option" datatype in the database. The content object attributes will be modified (converted) so that they use the new "Multi-option2" datatype.

The following table reveals the available options for this script.

Option	Description
--all-classes	The script will go through all types of content objects.
--contentclass=value	The script will only update objects of the specified class. You need to replace "value" with the actual identifier of the class.

Note: overridden templates of the public siteaccess for the "Multi-option" datatype will also need to be updated (manually).

Changes to roles and policies

From 3.10, users (typically those to which the anonymous role applies) need to have access to the "tipafriend" function of the "content" module in order to use the "Tip a friend" feature. Because of compatibility reasons, you need to run the "updatetipafriendpolicy.php" script.

Note: there is no need to run the script if you are upgrading from 3.9.3 or later versions (3.9.5 or later if you are running eZ Publish on Windows).

The following example shows how to run the script:

1. Navigate into the eZ Publish 3.10.1 directory.
2. Run the script using the following shell command:

```
php update/common/scripts/3.10/
updatetipafriendpolicy.php -s example_admin -l login -p password
```

Replace "example_admin" with the actual name of your admin siteaccess and use the administrator's login and password instead of "login" and "password".

The script will add a new role that grants access to the "Tip a friend" feature and ask for confirmation about assigning this role to each user or user group located one level beneath the "Users" top level node.

Improved handling of country dependent VATs

In eZ Publish version 3.10.0 - 3.10.1, the handling of country dependent VATs has been improved. Because of compatibility reasons, you need to run the "updatevatcountries.php" script.

Note: there is no need to run the script if you are upgrading from 3.9.5 or a later version.

The "updatevatcountries.php" script should be run once for each database, specifying one siteaccess per database. If you only have a public and an administration siteaccess that share the same database, you only need to run the script for one of the siteaccesses. If the siteaccess isn't specified, the default siteaccess will be used.

The following example shows how to run the script:

1. Navigate into the eZ Publish 3.10.1 directory.
2. Run the "updatevatcountries.php" script (replace "example" with the actual name of your siteaccess):

```
php update/common/scripts/3.10/updatevatcountries.php -s example
```

The script will go through all the VAT rules in the database and make them compatible with the newly introduced changes.

Binary files uploaded via WebDAV (optional)

In eZ Publish version 3.9.4 and earlier versions, there is a bug in the "File" datatype which may cause problems with binary file uploading via WebDAV. The typical symptoms are that the file extension of the uploaded file is missing and that you may not be able to download the file once it is uploaded (refer to <http://issues.ez.no/9450> for more information). The same bug exists in eZ Publish 3.10.0. This issue was fixed in eZ Publish 3.9.5, 3.10.1 and 4.0.0.

If your site has objects that were created by uploading files via WebDAV and you have configured MIME-type-to-class mapping using the "MimeClassMap[]" directive in an override for "upload.ini", you need to run the "updatebinaryfile.php" script in order to fix the file extensions. (If a content class does not have any attributes that make use of the "File" datatype, instances of this class are not affected.)

Note: there is no need to run the script if you are upgrading from 3.9.5 or a later version.

The "updatebinaryfile.php" script should be run once for each database, specifying one siteaccess per database. If you only have a public and an administration siteaccess that share the same database, you only need to run the script for one of the siteaccesses. If the siteaccess isn't specified, the default siteaccess will be used.

The following example shows how to run the script.

1. Navigate into the eZ Publish 3.10.1 directory.

2. Run the script (replace "example" with the actual name of your siteaccess):

```
php update/common/scripts/3.10/updatebinaryfile.php -s example
```

The script will go through all the binary files and fix their extensions. The filename entries stored in the "ezbinaryfile" database table will be changed accordingly.

Step 4: Updating the system configuration

Updating INI settings that use URL aliases

If your site makes use of the "urlalias" or "urlalias_iri" transformation method, read this part carefully. If you have specified "urlalias_compat" in the "TransformationGroup" configuration setting in step 3, skip this part.

After running the "updateniceurls.php" script in step 3, you need to review your configuration settings and update the URL aliases that are used in configuration files. For example, if your site makes use of the "PathPrefix" and "PathPrefixExclude" settings, this functionality might stop working if the specified URL aliases have changed (which will likely be the case because the virtual URLs are re-generated in accordance with the "TransformationGroup" and "WordSeparator" directives). Even though the old virtual URLs are stored as history entries and redirect to new ones when being accessed on the actual site, configuration settings do not work with URL history entries, i.e. you need to specify correct URL aliases there.

Also, you need to review the configuration settings in your "override.ini.append.php" file(s), otherwise your override templates activated by the "url_alias" condition will most likely stop working. For each line in the "override.ini.append.php" file that contains something like this:

```
Match[url_alias]=<virtual_url>
```

make sure that the correct virtual URL is specified.

Note that for multi-language sites, you need to specify URL aliases in the most prioritized language (controlled by the "SiteLanguageList" configuration array located under the "[RegionalSettings]" section of "site.ini.append.php").

Tree menu configuration

In eZ Publish versions prior to 3.10, the left tree menu in the administration interface didn't scale very well (refer to the [specifications](#) for details). This has been substantially improved in 3.10.0. The newly added implementation is called "dynamic content structure menu". It makes use of the [AJAX](#) technology, which is supported by all modern browsers. The old implementation is still available and enabled after upgrading. It is however recommended to switch to the new implementation, specially for large sites with many nodes. This will make the tree menu work much faster and decrease the usage of network bandwidth. (Note that new sites created from scratch using the setup wizard will have "dynamic" tree menu in the administration interface.)

Note that eZ Publish 3.10.1 introduces some changes to the dynamic tree menu functionality. In particular, the "treemenu" view of the "content" module no longer makes use of parameters that are transferred using GET variables (this improves tree menu compatibility when running PHP in FastCGI mode; refer to <http://issues.ez.no/11806> for more information).

If you want to use the new implementation of the tree menu, do the following:

1. Open the "contentstructuremenu.ini.append.php" configuration file located in the "settings/siteaccess/example_admin" directory (replace "example_admin" with the actual name of your admin siteaccess) and edit it. If the file does not exist, create it.
2. Add the following lines under the "[TreeMenu]" section:

```
[TreeMenu]
Dynamic=enabled
```

Updating Apache configuration (optional)

It is possible to boost the performance of the dynamic tree menu by using the "index_treemenu.php" script. This can be done by instructing Apache to use "index_treemenu.php" instead of "index.php" when the treemenu view of the content module is requested. This is an optimized script that only takes care about a specific case and thus it works faster than the regular "index.php".

Note that due to a bug in eZ Publish 3.10.1, you need to download the updated version of "index_treemenu.php" from <http://pubsvn.ez.no> and replace the corresponding file in your installation. The same bug exists in eZ Publish 4.0.1.

Then, update your ".htaccess" file to make the system use "index_treemenu.php" when needed. This can be done by adding the following lines above the existing rewrite rules for eZ Publish:

```
RewriteRule content/treemenu/? index_treemenu.php
RewriteRule index_treemenu.php - [L]
```

If no ".htaccess" file is used, it is strongly recommended to update the Apache configuration file. You can do this by adding the following line above the existing rewrite rules for eZ Publish:

```
RewriteRule content/treemenu/? /index_treemenu.php [L]
```

anchors within headings in Online Editor

eZ Publish 3.10 comes with the Online Editor extension version 4.2.4 and later. In these versions, the "AnchorAsAttribute" setting located in the [header] section of the "extension/ezdhtml/settings/content.ini.append" configuration file is disabled by default (refer to <http://issues.ez.no/11312> for more information). The above mentioned setting controls whether the usage of the deprecated "anchor_name" attribute for the "header" XML tag should be allowed

or not. With the new default configuration, the Online Editor will automatically convert all occurrences of this attribute into separate "anchor" tags when you edit a site page. If XML content on your site is using this attribute and you want to keep the old behavior, enable the "AnchorAsAttribute" setting from within the "settings/override/content.ini.append.php" configuration file. (Do not edit "extension/ezdhtml/settings/content.ini.append"; this file is overwritten whenever you upgrade the extension.)

Step 5: Clearing the caches

Whenever an eZ Publish solution is upgraded, all caches must be cleared in a proper way. This should be done from within a system shell:

1. Navigate into the eZ Publish 3.10 directory.
2. Run the script using the following shell command:

```
php bin/php/ezcache.php --clear-all --purge
```

Purging allows to make sure the caches are physically removed. Without the "--purge" option, the caches will be expired but not removed.

Sometimes the script is unable to clear caches because of restrictive file/directory permission settings. Make sure that all caches have been cleared by inspecting the contents of the various cache subdirectories within the "var" directory (typically "var/cache/" and "var/<name_of_siteaccess>/cache/" directories). If there are caches left, you need to remove them manually.

1.15.3 from 3.9.x to 3.10.0 (deprecated)

This documentation page is deprecated and no longer maintained. Refer to the following documentation pages for upgrade instructions:

- [Upgrading from 3.10.x to 3.10.y \(page 235\)](#)
- [Upgrading from 3.9.x to 3.10.y \(page 242\)](#)

This section describes how to upgrade your existing eZ Publish 3.9.x installation to version 3.10.0. If you are upgrading from a version prior to eZ Publish 3.9.0, you need to first upgrade to 3.9.0 (page [272](#)) before you can upgrade to 3.10.

The upgrade procedure described below is generic and does not cover any specific cases. If you are running eZ Publish in a clustered environment, refer to the [developer documentation](#) for upgrade instructions. In case the eZ Publish Extension for Oracle Database is used, refer to the documentation of the database extension.

Important note

Before continuing, note that eZ Publish 3.10.0 does not support wildcard based URL forwarding anymore. If your site makes use of this functionality, it is strongly recommended to delay upgrading until eZ Publish 3.10.1 is released.

Make sure that you have a working backup of the site before you do the actual upgrade. The upgrade procedure consists of the following steps:

1. Upgrading the distribution files to 3.10
2. Upgrading the database to 3.10
3. Running the 3.10 upgrade scripts
4. Updating the system configuration
5. Clearing the caches

Step 1: Upgrading the distribution files

The easiest way to upgrade the distribution files is to unpack eZ Publish 3.10 to a directory and then copy the directories that contain site-specific files from the existing installation. Make sure that you copy the following directories:

- design/example
- design/example_admin
- var

- settings/siteaccess
- settings/override

Replace "example" and "example_admin" with actual names used by your siteaccesses.

Important bug fix for the "user" module

In eZ Publish 3.10.0, the "register" view of the "user" module contains a bug. It prevents the system from registering new users when the "VerifyUserEmail" setting is enabled (it is enabled by default). The same bug exists in eZ Publish 3.8.10 and 3.9.4 (refer to <http://issues.ez.no/11598> for more information). This issue will be fixed in the upcoming eZ Publish 3.9.5 and 3.10.1 releases. In order to make your user registration mechanism work properly in 3.10.0, you have to download the updated version of "kernel/user/register.php" from <http://pubsvn.ez.no> and replace the corresponding file in your installation.

Custom extensions

If you are using custom extensions, the subdirectories inside the "extension" directory will also have to be copied. However, make sure that you do not overwrite any extensions that come with eZ Publish (for example the "PayPal" extension).

If you are using the Website Interface front-end, make sure you copied the "extension/ezwebin/" directory. It is also possible to upgrade the Website Interface extension to a newer version (you can find more information and instructions in the "Upgrading the Website Interface" chapter of the Website Interface installation guide).

Translations

eZ Publish 3.10.0 has less translations than 3.9 releases because not all of the translation contributors have submitted a CLA yet. If you are using a multilingual site the possible workarounds in that case can be either to use the 3.9.x translations or to download translations from <http://ez.no/developer/translations>. It would also be highly appreciated if translators for existing translations send us the CLA so that we can include the translations in future 3.10 releases.

Step 2: Upgrading the database

The following text describes how a 3.9.x database can be upgraded to 3.10.0.

Before reading on, note that some of the database changes from eZ Publish 3.9.0 to 3.10.0 have been included in other 3.9 releases. If you are upgrading from a version later than 3.9.0, it is recommended to skip some parts of the database upgrade script. These are marked with "from 3.9.z" comments in the ".sql" file; make sure you skip all sections with "z" less than or equal to the last digit in your current version number. For example, if you are upgrading from 3.9.3, skip sections marked with "from 3.9.1", "from 3.9.2" and "from 3.9.3" comments (if any).

MySQL

1. Navigate into the eZ Publish 3.10.0 directory.
2. Run the database upgrade script:

```
mysql -u <username> -p<password> <database> < update/database/mysql/3.10/  
dbupdate-3.9.0-to-3.10.0.sql
```

Note that the CREATE TABLE statements in the database upgrade script do not specify which storage engine to use (no ENGINE or TYPE option), and thus the default storage engine will be used. Normally, it is MyISAM (starting from MySQL v.3.23). If you are using InnoDB, make sure the default storage engine is set to InnoDB before you run the database upgrade script (refer to the [MySQL documentation](#) for information about how to set the default engine). If you were not able to change the MySQL configuration on your server, and the upgrade left you with a mix of table types, you can convert the newly created tables to InnoDB using the following SQL queries:

```
ALTER TABLE ezisbn_group TYPE = innodb;  
ALTER TABLE ezisbn_group_range TYPE = innodb;  
ALTER TABLE ezisbn_registrant_range TYPE = innodb;  
ALTER TABLE ezurlalias_ml TYPE = innodb;
```

It is also possible to use the "bin/php/ezconvertmysqltabletype.php" script for database conversion.

PostgreSQL

1. Navigate into the eZ Publish 3.10.0 directory.
2. Run the database upgrade script:

```
psql -d <database> -U <downer> < update/database/postgresql/3.10/  
dbupdate-3.9.0-to-3.10.0.sql
```

Step 3: Running the system upgrade scripts

The 3.10.0 version of eZ Publish introduces a couple of new features. In order to make sure that your site is compatible with these features, you'll have to run a couple of upgrade scripts.

Multi-language support for URL aliases

In eZ Publish 3.10, a new feature has been added that makes it possible to use multi-lingual virtual URLs (also known as nice URLs or URL aliases). In order to use this new functionality, you will have to specify the desired type of URL transformation using the "Transformation-Group" setting located in the [URLTranslator] section of an override for "site.ini" and run the "updateniceurls.php" script as described below.

Note that this script has been sufficiently improved in eZ Publish 4.0.0 and the upcoming version 3.10.1. In addition, these versions of eZ Publish will support wildcard based URL forwarding (This feature was removed when implementing the multilingual URLs functionality for eZ Publish 3.10.0 and then re-added in later versions.) If you are planning to upgrade from 3.9.x to 4.0.0, it is recommended to skip running the "updateniceurls.php" script when upgrading to 3.10.0, and run the updated version of the script at a later stage (when upgrading from 3.10.0 to 4.0.0). If you are going to use multilingual URLs in eZ Publish 3.10, it is recommended to delay upgrading until the upcoming 3.10.1 version is out.

It is also possible to choose which word separator should be used in the URLs. This can be done by setting the "WordSeparator" configuration directive also located in the [URLTranslator] section to either "dash", "underscore" or "space". Note that this setting will be ignored when the "urlalias_compat" transformation is used (it only supports underscores as separators).

Before running the script, make sure that you have specified the same transformation settings for all siteaccesses that are using the same database. If all your siteaccesses are using the same database (which is the most typical/usual case), you can set the "TransformationGroup" and "WordSeparator" directives in "settings/override/site.ini.append.php". Otherwise, you will have to configure URL transformation from within "settings/siteaccess/example/site.ini.append.php" (where "example" should be replaced by the actual name of your siteaccess) for each siteaccess.

The "updateniceurls.php" script should be run once for each database, specifying one siteaccess per database. If you only have a public and an administration siteaccess that share the same database, you only need to run the script for one of the siteaccesses. If the siteaccess is not specified, the default siteaccess will be used.

The following example shows how to run the script.

1. Navigate into the eZ Publish 3.10.0 directory.
2. Run the script (replace "example" with the actual name of your siteaccess):

```
php bin/php/updateniceurls.php -s example
```

The script will go through all content objects stored in the database and create new virtual URLs for them in accordance with the specified transformation settings. Old virtual URLs will be transferred to the new system and stored as history entries. Note that new URL aliases will be generated for all the languages of your site regardless if they are available or not for the specified siteaccess. The time it takes to update all URL aliases is proportional to the number of nodes on your site. Newly generated URL aliases are stored in a database table called "ezurlalias_ml".

Refer to "Multi-language support for URL aliases" for more information about this feature.

ISBN datatype enhancements

In eZ Publish 3.10, the handling of ISBN-13 numbers has been significantly improved (refer to the ISBN datatype documentation for details). The different ranges are now stored in the database, allowing the system to fix up incorrectly entered ISBN-13 codes (codes with misplaced or missing

dashes). When a new version of eZ Publish is released, the latest information about the different ranges is downloaded from <http://www.isbn-international.org/converter/ranges.js> and stored in a ".dba" file. During the installation process, the setup wizard imports this information to the database. When upgrading, you need to import ranges to the database manually. This must be done even though your site does not use ISBN-13 numbers. You can either import ranges from the ".dba" file using the "ezimportdbafile.php" script or download the latest "range.js" file and run the "updateisbn13.php" script.

The following example shows how to import ranges from the ".dba" file:

1. Navigate into the eZ Publish 3.10.0 directory.
2. Run the script (replace "example" with the actual name of your siteaccess):

```
php bin/php/ezimportdbafile.php --datatype=eziisbn -s example
```

The "datatype" parameter tells the script which datatype the data should be imported for.

The next example shows how to get the latest ranges from the ISBN International website.

Note that due to a [bug](#) in eZ Publish 3.10.0, you need to download the updated version of the "updateisbn13.php" script from <http://pubsvn.ez.no> and replace the corresponding file in your installation before running the script. The updated version of the script is included in eZ Publish 4.0.0 and the upcoming 3.10.1 release.

1. Download a ".js" file containing the list of ranges using the following link: <http://www.isbn-international.org/converter/ranges.js>
2. Navigate into the eZ Publish 3.10.0 directory and copy the downloaded file there.
3. Run the "updateisbn13.php" script using the following shell command:

```
php bin/php/updateisbn13.php --file=ranges.js
```

The script will import information from the "range.js" file to the database.

In order to use ISBN-13 numbers on your site after upgrading, you need to do the following:

1. Make sure that the "ISBN-13 format" checkbox in the class attribute edit interface for the corresponding attribute is enabled (it is off by default).
2. Navigate into the eZ Publish 3.10.0 directory.
3. Run the "ezconvert2isbn13.php" script (replace "example" with the actual name of your siteaccess):

```
php bin/php/ezconvert2isbn13.php -s example
```

The script will go through all content object attributes using the "ISBN" datatype in the database and convert their values into valid ISBN-13 numbers. The following table reveals the available options for this script.

Option	Description
--class-id=value	The script will only update attributes of content objects that belong to the specified class. You need to replace "value" with the actual ID number of the class.
--attribute-id=value	The script will only update content object attributes that correspond to the specified content class attribute. You need to replace "value" with the actual ID number of the class attribute.
--all-classes	The script will go through all content objects.
--force	The script will update all attributes of the "ISBN" datatype regardless if the "ISBN-13 format" checkbox is enabled for the class attribute. It will also enable this setting at the class attribute level where it was previously disabled.

These scripts should be run once for each database, specifying one siteaccess per database. If you only have a public and an administration siteaccess that share the same database, you only need to run the scripts for one of the siteaccesses. If the siteaccess is not specified, the default siteaccess will be used.

Enhanced Multi-option datatype

In eZ Publish 3.10, a new datatype called "Multi-option2" has been added in order to replace the old "Multi-option" datatype. The new datatype allows option selections and makes it possible to create multiple and distinctive groups of multi-options for each content object. In addition, it supports rules and provides a handful of new and interesting features. Note that the old "Multi-option" datatype is deprecated and should no longer be used (it will be removed in the future).

In order to be compatible with the new datatype, you will have to run the "updatemultioption.php" script. This script should be run once for each database, specifying one siteaccess per database. If you only have a public and an administration siteaccess that share the same database, you only need to run the script for one of the siteaccesses. If the siteaccess is not specified, the default siteaccess will be used.

The following example shows how to run the script:

1. Navigate into the eZ Publish 3.10.0 directory.
2. Run the script (replace "example" with the actual name of your siteaccess):

```
php update/common/scripts/3.10/updatemultioption.php -s example
```

The script will go through all content object attributes that are using the old "Multi-option" datatype in the database. The content object attributes will be modified (converted) so that they use the new "Multi-option2" datatype.

The following table reveals the available options for this script.

Option	Description
--all-classes	The script will go through all types of content objects.
--contentclass=value	The script will only update objects of the specified class. You need to replace "value" with the actual identifier of the class.

Note: overridden templates of the public siteaccess for the "Multi-option" datatype will also need to be updated (manually).

Changes to roles and policies

In eZ Publish 3.9.3 and later versions, users (typically those to which the anonymous role applies) need to have access to the "tipafriend" function of the "content" module in order to use the "Tip a friend" feature. If you are upgrading from a version prior to 3.9.3, you need to run the "updatetipafriendpolicy.php" script as described below.

If you are running eZ Publish on Windows, note that in eZ Publish 3.10.0 there is a bug in the ?updatetipafriendpolicy.php? script that prevents it from granting access to the "Tip a friend" feature to the users. You need to download the updated version of the "updatetipafriendpolicy.php" script from <http://pubsvn.ez.no> and replace the corresponding file in your installation before running the script. The same bug exists in eZ Publish 3.9.3 - 3.9.4 (refer to <http://issues.ez.no/11663> for more information). The updated version of the script is included in eZ Publish 4.0.0 and the upcoming eZ Publish 3.10.1 and 3.9.5 releases.

The following example shows how to run the script:

1. Navigate into the eZ Publish 3.10.0 directory.
2. Run the script using the following shell command:

```
php update/common/scripts/3.10/updatetipafriendpolicy.php -s example_admin -l login -p password
```

Replace "example_admin" with the actual name of your admin siteaccess and use the administrator's login and password instead of "login" and "password".

The script will add a new role that grants access to the "Tip a friend" feature and ask for confirmation about assigning this role to each user or user group located one level beneath the "Users" top level node.

Improved handling of country dependent VATs

In eZ Publish version 3.10, the handling of country dependent VATs has been improved. Because of compatibility reasons, you need to run the "updatevatcountries.php" script.

Note that due to a bug in eZ Publish 3.10.0, you need to download the updated version of the "updatevatcountries.php" script from <http://pubsvn.ez.no> and replace the corresponding file in your installation before running the script. The same bug exists in eZ Publish 3.9.4 and 3.9.3 (refer to <http://issues.ez.no/11955> for more information). The updated version of the script is included in eZ Publish 4.0.0 and the upcoming eZ Publish 3.10.1 and 3.9.5 releases.

The "updatevatcountries.php" script should be run once for each database, specifying one siteaccess per database. If you only have a public and an administration siteaccess that share the same database, you only need to run the script for one of the siteaccesses. If the siteaccess isn't specified, the default siteaccess will be used.

The following example shows how to run the script:

1. Navigate into the eZ Publish 3.10.0 directory.
2. Run the "updatevatcountries.php" script (replace "example" with the actual name of your siteaccess):

```
php update/common/scripts/3.10/updatevatcountries.php -s example
```

The script will go through all the VAT rules in the database and make them compatible with the newly introduced changes.

Step 4: Updating the system configuration

Updating INI settings that use URL aliases

If your site makes use of the "urlalias" or "urlalias_iri" transformation method, read this part carefully. If you have specified "urlalias_compat" in the "TransformationGroup" configuration setting in step 3, skip this part.

After running the "updateniceurls.php" script in step 3, you need to review your configuration settings and update the URL aliases that are used in configuration files. For example, if your site makes use of the "PathPrefix" and "PathPrefixExclude" settings, this functionality might stop working if the specified URL aliases have changed (which will likely be the case because the virtual URLs are re-generated in accordance with the "TransformationGroup" and "WordSeparator" directives). Even though the old virtual URLs are stored as history entries and redirect to new ones when being accessed on the actual site, configuration settings do not work with URL history entries, i.e. you need to specify correct URL aliases there.

Also, you need to review the configuration settings in your "override.ini.append.php" file(s), otherwise your override templates activated by the "url_alias" condition will most likely stop working. For each line in the "override.ini.append.php" file that contains something like this:

```
Match[url_alias]=<virtual_url>
```

make sure that the correct virtual URL is specified.

Note that for multi-language sites, you need to specify URL aliases in the most prioritized language (controlled by the "SiteLanguageList" configuration array located under the "[RegionalSettings]" section of "site.ini.append.php").

Treemenu configuration

In eZ Publish versions prior to 3.10, the left treemenu in the administration interface didn't scale very well (refer to the [specifications](#) for details). This has been significantly improved in 3.10. The newly added implementation is called "dynamic content structure menu". It makes use of the [AJAX](#) technology, which is supported by all modern browsers. The old implementation is still available and enabled after upgrading. It is however recommended to switch to the new implementation, specially for large sites with many nodes. This will make the treemenu work much faster and decrease the usage of network bandwidth. (Note that new sites created from scratch using the setup wizard will have "dynamic" treemenu in the administration interface.)

If you want to use the new implementation of the treemenu, do the following:

1. Open the "contentstructuremenu.ini.append.php" configuration file located in the "settings/siteaccess/example_admin" directory (replace "example_admin" with the actual name of your admin siteaccess) and edit it. If the file does not exist, create it.
2. Add the following lines under the "[TreeMenu]" section:

```
[TreeMenu]
Dynamic=enabled
```

Updating Apache configuration (optional)

The following rewrite rules are added into ".htaccess_root" file by default.

```
RewriteRule content/treemenu/?$ index_treemenu.php
RewriteRule index_treemenu.php - [L]
```

If the ".htaccess_root" file isn't used, it is strongly recommended to update the Apache configuration file. You can do this by adding the following line above the other / already existing rewrite rules for eZ Publish:

```
RewriteRule content/treemenu/?$ /index_treemenu.php [L]
```

This will instruct Apache to use "index_treemenu.php" instead of "index.php" when accessing the "treemenu" view of the "content" module (this is an optimized script that only takes care about a specific case and thus it works faster than the regular "index.php").

Anchors within headings in Online Editor

eZ Publish 3.10.0 comes with 4.2.4 of the Online Editor extension. In this version, the "AnchorAsAttribute" setting located in the [header] section of the "extension/ezdhtml/settings/content.ini.append" configuration file is disabled by default. The above mentioned setting controls whether the usage of the "anchor_name" attribute for the "header" XML tag should be allowed or not. If XML content on your site is using this attribute, you need to make sure that the "AnchorAsAttribute" setting is enabled.

Step 5: Clearing the caches

Whenever an eZ Publish solution is upgraded, all caches must be cleared in a proper way. This should be done from within a system shell:

1. Navigate into the eZ Publish 3.10 directory.
2. Run the script using the following shell command:

```
php bin/php/ezcache.php --clear-all --purge
```

Purging allows to make sure the caches are physically removed. Without the "--purge" option, the caches will be expired but not removed.

Sometimes the script is unable to clear caches because of restrictive file/directory permission settings. Make sure that all caches have been cleared by inspecting the contents of the various cache subdirectories within the "var" directory (typically "var/cache/" and "var/<name_of_siteaccess>/cache/" directories). If there are caches left, you need to remove them manually.

1.16 Upgrading to 3.9

This chapter contains information related to the upgrade of an eZ Publish solution to [version 3.9](#). Please follow the upgrade procedures that are described in the following sections:

- [Upgrading from 3.9.x to 3.9.y \(page 265\)](#)
- [Upgrading from 3.8.x to 3.9.0 \(page 272\)](#)

1.16.1 from 3.9.x to 3.9.y

This section describes how to upgrade your existing eZ Publish 3.9.x installation to version 3.9.y, for example from 3.9.0 to 3.9.5. If you are upgrading from a version prior to eZ Publish 3.9.0, you should first upgrade to 3.9.0 as described in this section (page 272).

Important note

Before continuing, note that it is strongly recommended to upgrade directly to the latest stable release in the 3.9 branch (3.9.5 at the time of writing), which contains all important bug fixes. Refer to the [changelogs](#) and [security advisories](#) for more information about the issues that were fixed in the latest releases or view the short list of changes below.

- There is a significant problem with eZ Publish 3.9.0 - 3.9.2. More precisely, the discount functionality of the shop module contains a security vulnerability. The same bug exists in eZ Publish 3.8.0 - 3.8.8. This issue was fixed in eZ Publish 3.8.9, 3.9.3 and 3.10.0. Refer to the [announcement](#) for details.
- eZ Publish versions 3.9.0 - 3.9.3 contain a security vulnerability that can be used to show hidden content on a public site access. The same bug exists in eZ Publish 3.8.0 - 3.8.9. This issue was fixed in eZ Publish 3.8.10, 3.9.4 and 3.10.0. Refer to the [announcement](#) for details.
- The "updatetipafriendpolicy.php" upgrade script in eZ Publish 3.9.3 - 3.9.4 contains a bug. The same bug exists in eZ Publish 3.10.0. It prevents the script from granting access to the "Tip a friend" feature to users. The problem occurs when the script is run on Windows. This issue was fixed in eZ Publish 3.9.5, 3.10.1 and 4.0.0. Refer to <http://issues.ez.no/11663> for more information.
- The "updatevatcountries.php" upgrade script in eZ Publish 3.9.3 - 3.9.4 contains a bug. The same bug exists in eZ Publish 3.10.0. This issue was fixed in eZ Publish 3.9.5, 3.10.1 and 4.0.0. Refer to <http://issues.ez.no/11955> for more information.
- In eZ Publish 3.9.0 - 3.9.4, there is a bug in the "File" datatype which may cause problems with binary file uploading via WebDAV. The same bug exists in eZ Publish 3.10.0. This issue was fixed in eZ Publish 3.9.5, 3.10.1 and 4.0.0. Refer to <http://issues.ez.no/9450> for more information.
- In eZ Publish 3.9.4, some cache related settings in the "settings/site.ini" configuration file were changed by mistake. This issue was fixed in eZ Publish 3.9.5. Refer to <http://issues.ez.no/11647> for more information.
- In eZ Publish 3.9.4, the "register" view of the "user" module contains a bug. It prevents the system from registering new users when the "VerifyUserEmail" setting is enabled (it is enabled by default). The same bug exists in eZ Publish 3.8.10 and 3.10.0. This issue was fixed in eZ Publish 3.9.5 and 3.10.1. Refer to <http://issues.ez.no/11598> for more information.

Make sure that you have a working backup of the site before you do the actual upgrade. The upgrade procedure consists of the following steps:

1. Upgrading the distribution files to 3.9.5
2. Upgrading the database to 3.9.5
3. Running the system upgrade scripts
4. Updating the system configuration
5. Clearing the caches

Step 1: Upgrading the distribution files

The easiest way to upgrade the distribution files is to unpack eZ Publish 3.9.5 to a directory and then copy the directories that contain site-specific files from the existing installation. Make sure that you copy the following directories:

- design/example
- design/example_admin
- var
- settings/siteaccess
- settings/override

Replace "example" and "example_admin" with actual names used by your siteaccesses.

Custom extensions

If you are using custom extensions then the subdirectories inside the "extension" directory will also have to be copied. However, make sure that you do not overwrite any extensions that come with eZ Publish (for example the "PayPal" extension).

If you are using the Website Interface front-end, make sure you copied the "extension/ezwebin/" directory. It is also possible to upgrade the Website Interface extension to a newer version (you can find more information and instructions in the "Upgrading the Website Interface" chapter of the Website Interface installation guide).

Important bug fix for remote IDs

Due to some bugs in previous versions of eZ Publish (3.9.0 and others), it might happen that not all content objects in your database have unique remote IDs. This basically means that the "remote_id" column of the "ezcontentobject" table in the database might contain duplicate values.

In eZ Publish 3.9.5, a database schema change was introduced in order to ensure that only unique remote IDs can be stored in this table. Because of this, you need to make sure that there are no content objects with identical remote IDs before upgrading the database. This can be done by running the "fixobjectremoteid.php" upgrade script. (Note that both the database schema change and the upgrade script were added to eZ Publish 3.10.1 and 4.0.1 as well.)

This script should be run once for each database, specifying one siteaccess per database. If you only have a public and an administration siteaccess that share the same database (which is the most typical/usual case), you only need to run the script for one of the siteaccesses. If the siteaccess isn't specified, the default siteaccess will be used.

The following example shows how to run the script:

1. Navigate into the eZ Publish 3.9.5 directory.
2. Run the script (replace "example" with the actual name of your siteaccess):

```
php update/common/scripts/3.9/fixobjectremoteid.php -s example
```

The script will search for objects with non-unique remote IDs. Every time such an object is found, the script will suggest to either display more detailed information or fix the problem automatically. If you choose the first option, the script will display the list of objects that have the same remote ID and let you decide which one should remain unchanged; all other objects in the list will get new remote IDs. Otherwise, the script will fix the problem automatically based on the objects' creation dates. This means that the script will generate new remote IDs for all objects in the list except the one that was created first.

The optional "--mode" parameter makes it possible to run the script in either "auto-fix" or "manual-fix" mode as shown below.

```
php update/common/scripts/3.9/fixobjectremoteid.php -s example --mode=value
```

If you replace "value" with "a" in the command above, the script will automatically fix all non-unique remote IDs that are found in the database based on the objects' creation dates. In this case, no further input is required from the user. If you specify "--mode=d", the script will display the list of objects for each non-unique remote ID so that you can manually choose which objects should keep their remote IDs.

Step 2: Upgrading the database

To upgrade a 3.9.0 database to 3.9.5, you should navigate into the eZ Publish 3.9.5 directory and run the following database upgrade scripts one after another:

1. dbupdate-3.9.0-to-3.9.1.sql
2. dbupdate-3.9.1-to-3.9.2.sql

3. dbupdate-3.9.2-to-3.9.3.sql
4. dbupdate-3.9.3-to-3.9.4.sql
5. dbupdate-3.9.4-to-3.9.5.sql

MySQL

The database upgrade scripts are located in the "update/database/mysql/3.9/" directory of your eZ Publish installation. Each of these scripts can be launched using the following shell command:

```
mysql -u <username> -p<password> <database> < update/database/mysql/3.9/  
dbupdate-3.9.x-to-3.9.y.sql
```

PostgreSQL

The database upgrade scripts are located in the "update/database/postgresql/3.9/" directory of your eZ Publish installation. Each of these scripts can be launched using the following shell command:

```
psql -d <database> -U <dbowner> < update/database/postgresql/3.9/  
dbupdate-3.9.x-to-3.9.y.sql
```

Step 3: Running the system upgrade scripts

The 3.9.1 - 3.9.5 versions of eZ Publish introduce a couple of important bug fixes and functionality changes. In order to make sure that your site is compatible with these changes, you may need to run a few upgrade scripts.

Changes to roles and policies

From 3.9.3, users (typically those to which the anonymous role applies) need to have access to the "tipafriend" function of the "content" module in order to use the "Tip a friend" feature. Because of compatibility reasons, you need to run the "updatetipafriendpolicy.php" script.

Note for Linux/UNIX users: there is no need to run the script if you previously upgraded to 3.8.9 or a later version in the 3.8.x branch.

Note for Windows users: in eZ Publish 3.9.3 - 3.9.4 the "updatetipafriendpolicy.php" script contains a bug that prevents it from granting access to the "Tip a friend" feature to the users. The same bug exists in eZ Publish 3.8.9 - 3.8.10 and 3.10.0 (refer to <http://issues.ez.no/11663> for more information). The updated version of the script is available in eZ Publish 3.9.5, 3.10.1 and 4.0.0. Windows users that previously upgraded to 3.8.9 - 3.8.10 or 3.9.3 - 3.9.4 using the original version of the script are encouraged to re-run this script when upgrading to 3.9.5.

The following example shows how to run the script:

1. Navigate into the eZ Publish 3.9.5 directory.
2. Run the script using the following shell command:

```
php update/common/scripts/3.9/  
updatetipafriendpolicy.php -s example_admin -l login -p password
```

Replace "example_admin" with the actual name of your admin siteaccess, use the administrator's login and password instead of "login" and "password".

The script will add a new role that grants access to the "Tip a friend" feature and ask for confirmation about assigning this role to each user or user group located one level beneath the "Users" top level node.

Improved handling of country dependent VATs

In eZ Publish version 3.9.3 - 3.9.5, the handling of country dependent VATs has been improved. In order to make sure that your site is compatible with the changes, you need to run the "updatevatcountries.php" script.

Note that the "updatevatcountries.php" upgrade script in eZ Publish 3.9.3 - 3.9.4 contains a bug. The same bug exists in eZ Publish 3.10.0 (refer to <http://issues.ez.no/11955> for more information). The updated version of the script is available in eZ Publish 3.9.5, 3.10.1 and 4.0.0. If you previously upgraded to 3.9.3 or 3.9.4 using the original version of the script, you need to run the updated version of the script when upgrading to 3.9.5.

The "updatevatcountries.php" script should be run once for each database, specifying one siteaccess per database. If you only have a public and an administration siteaccess that share the same database, you only need to run the script for one of the siteaccesses. If the siteaccess isn't specified, the default siteaccess will be used.

The following example shows how to run the script:

1. Navigate into the eZ Publish 3.9.5 directory.
2. Run the "updatevatcountries.php" script (replace "example" with the actual name of your siteaccess):

```
php update/common/scripts/3.9/updatevatcountries.php -s example
```

The script will go through all the VAT rules in the database and make them compatible with the newly introduced changes.

Binary files uploaded via WebDAV (optional)

In eZ Publish 3.9.0 - 3.9.4, there is a bug in the "File" datatype which may cause problems with binary file uploading via WebDAV. The typical symptoms are that the file extension of the

uploaded file is missing and that you may not be able to download the file once it is uploaded (refer to <http://issues.ez.no/9450> for more information). The same bug exists in eZ Publish 3.8.6 - 3.8.10 and 3.10.0. This issue was fixed in eZ Publish 3.9.5, 3.10.1 and 4.0.0.

If your site has objects that were created by uploading files via WebDAV and you have configured MIME-type-to-class mapping using the "MimeClassMap[]" directive in an override for "upload.ini", you need to run the "updatebinaryfile.php" script in order to fix the file extensions. (If a content class does not have any attributes that make use of the "File" datatype, instances of this class are not affected.)

The "updatebinaryfile.php" script should be run once for each database, specifying one siteaccess per database. If you only have a public and an administration siteaccess that share the same database, you only need to run the script for one of the siteaccesses. If the siteaccess isn't specified, the default siteaccess will be used.

The following example shows how to run the script:

1. Navigate into the eZ Publish 3.9.5 directory.
2. Run the script (replace "example" with the actual name of your siteaccess):

```
php update/common/scripts/3.9/updatebinaryfile.php -s example
```

The script will go through all the binary files and fix their extensions. The filename entries stored in the "ezbinaryfile" database table will be changed accordingly.

Step 4: Updating the system configuration

You are not required to do any configuration changes when upgrading from 3.9.0 to 3.9.5.

Default configuration of caches (optional)

In eZ Publish versions prior to 3.9.4, the default configuration enables view caching, template caching and template compiling. (This configuration of caches is recommended for live sites. During development, these features can be turned off by disabling the "ViewCaching", "TemplateCache" and "TemplateCompile" directives within an override for "site.ini".) However, this default configuration was accidentally changed in eZ Publish 3.9.4 (refer to <http://issues.ez.no/11647> for more information). The change is reverted in eZ Publish 3.9.5.

Default configuration of unique user emails (optional)

In eZ Publish versions prior to 3.8.0, the default configuration does not allow different users to be registered with the exact same email address. This is just a built-in precaution mechanism which can be easily turned off by setting the "RequireUniqueEmail" directive within the [UserSettings] block of a configuration override for "site.ini" to "false". However, this default configuration was accidentally changed in eZ Publish version 3.8.0 (refer to <http://issues.ez.no/9643> for more information). The change is reverted in eZ Publish versions 3.8.7 and 3.9.1.

Step 5: Clearing the caches

Whenever an eZ Publish solution is upgraded, all caches must be cleared in a proper way. This should be done from within a system shell:

1. Navigate into the eZ Publish 3.9.5 directory.
2. Run the clear cache script:

```
bin/shell/clearcache.sh --clear-all
```

Sometimes the script is unable to clear all cache files because of restrictive file/directory permission settings. Make sure that all cache files have been cleared by inspecting the contents of the various cache subdirectories within the "var" directory (typically "var/cache/" and "var/<name_of_siteaccess>/cache/" directories). If there are any cache files left, you need to remove them manually.

1.16.2 from 3.8.x to 3.9.0

This section describes how to upgrade your existing eZ Publish 3.8.x installation to version 3.9.0. If you are upgrading from a version prior to eZ Publish 3.8.0, you need to first upgrade to 3.8.0 before you can upgrade to 3.9.0.

Important note

Before continuing, note that it is recommended to upgrade directly to the latest stable release in the 3.9 branch. (Refer to "Upgrading from 3.a.b to 3.x.y (page 350)" and "Upgrading from 3.9.x to 3.9.y (page 265)" to learn how this can be done.) Mainly, there are two reasons for this:

- The "correctxmltext.php" upgrade script in eZ Publish 3.9.0 contains a bug. It makes the script crash. This issue was fixed in 3.9.1. Refer to <http://issues.ez.no/9952> for more information.
- There is a significant problem with eZ Publish 3.9.0-3.9.2. More precisely, the discount functionality of the shop module contains a security vulnerability. The issue has been fixed in version 3.9.3. Refer to the [announcement](#) and the upgrade instructions for details.

Overview

Make sure that you have a working backup of the site before you do the actual upgrade. The upgrade procedure consists of the following steps:

1. Upgrading the distribution files to 3.9.0
2. Upgrading the database to 3.9.0
3. Running the 3.9.0 upgrade scripts
4. Updating the system configuration
5. Clearing the caches

Step 1: Upgrading the distribution files

The easiest way to upgrade the distribution files is to unpack eZ Publish 3.9.0 to a directory and then copy the directories that contain site-specific files from the existing installation. Make sure that you copy the following directories:

- design/example
- design/example_admin
- var

- settings/siteaccess
- settings/override

Replace "example" and "example_admin" with actual names used by your siteaccesses.

Custom extensions

If you are using custom extensions, the subdirectories inside the "extension" directory will also have to be copied. However, make sure that you do not overwrite any extensions that come with eZ Publish (for example the "PayPal" extension).

Step 2: Upgrading the database

The following text describes how a 3.8.0 database can be upgraded to 3.9.0. Before reading on, note that some of the database changes from eZ Publish 3.8.0 to 3.9.0 have been included in eZ Publish versions 3.8.1 and 3.8.5 as well. If you are upgrading from a version later than 3.8.0, it is recommended to skip some parts of the database upgrade scripts (the sections that should be skipped are marked with "from 3.8.1" or "from 3.8.5" comments in the ".sql" file).

MySQL

1. Navigate into the eZ Publish 3.9.0 directory.
2. Run the database upgrade script:

```
mysql -u <username> -p<password> <database> < update/database/mysql/3.9/  
dbupdate-3.8.0-to-3.9.0.sql
```

Note that the CREATE TABLE statements in the database upgrade script do not specify which storage engine to use (no ENGINE or TYPE option), and thus the default storage engine will be used. Normally, it is MyISAM (starting from MySQL v.3.23). If you are using InnoDB, make sure the default storage engine is set to InnoDB before you run the database upgrade script (refer to [MySQL documentation](#) for information about how to set the default engine). If you were not able to change the MySQL configuration on your server, and the upgrade left you with a mix of table types, you can convert the newly created tables to InnoDB using the following SQL queries:

```
ALTER TABLE ezcontentobject_trash TYPE = innodb;  
ALTER TABLE ezcontentclass_name TYPE = innodb;  
ALTER TABLE ezsearch_search_phrase TYPE = innodb;  
ALTER TABLE ezsearch_return_count TYPE = innodb;
```

It is also possible to use the "bin/php/ezconvertmysqltabletype.php" script for database conversion.

PostgreSQL

1. Navigate into the eZ Publish 3.9.0 directory.
2. Run the database upgrade script:

```
psql -d <database> -U <downer> < update/database/postgresql/3.9/  
dbupdate-3.8.0-to-3.9.0.sql
```

Step 3: Running the 3.9.0 upgrade scripts

The 3.9.0 version of eZ Publish introduces a couple of new features. In order to make sure that your site is compatible with these features, you'll have to run a couple of upgrade scripts.

Translating names of classes and class attributes

In eZ Publish 3.9, a new feature has been added to the multi-language functionality that makes it possible to translate names of the eZ Publish content classes / class attributes (see [this page](#) for more information). In order to make sure that your site is compatible with this new functionality, you will have to run the "updateclasstranslations.php" script. This step must be done even though your site has only one language.

The "updateclasstranslations.php" script should be run for all siteaccesses that use different databases. It is strongly recommended that you only run the script once per database, otherwise your class and attribute names may be corrupted. If you only have a public and an administration siteaccess (which is the most typical/usual case), you will only need to run the script for one of the siteaccesses.

The "language" parameter tells the script which language your class / class attribute names currently exist in. Note that only one language can be specified using this parameter. If your classes are in different languages, it is recommended to specify the language that is used for most of them (for example, if 10 default class names are in English and 2 names of newly created custom classes are in Norwegian, you should use English as a parameter for the script and then edit the two custom classes manually). The following example shows how to run the script.

1. Navigate into the eZ Publish 3.9.0 directory.
2. Run the script (replace "example" with the actual name of your siteaccess, specify the original language instead of "eng-GB"):

```
php update/common/scripts/3.9/  
updateclasstranslations.php -s example --language=eng-GB
```

The script will check your content classes and content class attributes and re-initialize their names for missing translations.

Custom attributes for XML tags

eZ Publish 3.9.0 supports using custom attributes in all XML tags (see [this page](#) for more information). The names of the custom attributes must be added to the global (or siteaccess) "content.ini.append.php" configuration file. For example, if you wish to use a specific summary field in your tables, you can add a custom attribute called "summary" to the "table" XML tag by adding the following lines to the [table] section of your "content.ini.append.php" file:

```
CustomAttributes []
CustomAttributes []=summary
```

In order to be compatible with this new feature, you will have to run the "correctxmltext.php" script. The following example shows how this can be done.

1. Navigate into the eZ Publish 3.9.0 directory.
2. If all your siteaccesses are using the same database, run the script with the "--global" option so that it will only update the "settings/override/content.ini.append.php" file:

```
php update/common/scripts/3.9/correctxmltext.php --global
```

Otherwise, run the script for each of your siteaccesses (replace "example" with the actual name of your siteaccess):

```
php update/common/scripts/3.9/correctxmltext.php -s example
```

Use "--help" to view the additional options supported by this script.

Note that this script is faulty. Refer to <http://issues.ez.no/9952> for more information about the problem and the solution.

The script will go through the XML block attributes stored in the database and do the following:

- Find all occurrences of the "object" tags (if any) and convert these to "embed" tags. (The "object" tag is deprecated and no longer used.)
- Check which custom classes are used within your XML tags and add missing "AvailableClasses[]" settings to your configuration file. (From 3.8.1, the names of all custom classes that are used within XML tags must be specified from within an override of the "content.ini" configuration file.)
- Check which custom attributes are used in your XML tags and add missing "CustomAttributes[]" settings (if any) to your configuration file.

Note that you can skip some of these tasks by using one of the options listed below:

Option	Description
--skip-objects	Skip searching for deprecated "object" tags

	and converting them.
--skip-classes	Skip checking custom classes and modifying the "AvailableClasses[]" settings.
--skip-custom	Skip checking custom attributes and updating the "CustomAttributes[]" settings.

If you do not wish to update your configuration file automatically, run the script like this:

```
php update/common/scripts/3.9/
correctxmltext.php --classes-dump-only --custom-dump-only
```

The "--classes-dump-only" option allows you to view the list of missing classes instead of having your "AvailableClasses[]" settings automatically updated. The "--custom-dump-only" option allows to view the list of missing custom attributes without modifying the "CustomAttributes[]" settings.

Enhanced object relations

In eZ Publish 3.9.0, the object relations functionality has been improved (see [this page](#) for more information). The system will now use four types of object relations called "common", "XML linked", "XML embedded" and "by attribute". The first three types are relations at the object level while the last one is at the attribute level. The "XML embedded" type has been added in order to distinguish object relations created by inserting an embedded object in the XML block attribute from relations created by adding an object to the list of related objects (this type of relations is now called "common"). The "XML linked" type is used for new relations that are automatically created by the system when an internal link is inserted in an attribute of the XML block datatype. Note that "XML linked" relations are automatically removed when corresponding internal links are deleted. Same goes for "XML embedded" relations and their "embed" tags.

In order to be compatible with this new feature, you will have to run the "updatetypedrelation.php" script. This script should be run for all siteaccesses that use different databases. If you only have a public and an administration siteaccess (which is the most typical/usual case), you will only need to run the script for one of the siteaccesses. If the siteaccess is not specified, the default siteaccess will be used.

The following text shows how this can be done.

1. Navigate into the eZ Publish 3.9.0 directory.
2. Run the script (replace "example" with the actual name of your siteaccess):

```
php update/common/scripts/3.9/updatetypedrelation.php -s example
```

The script will go through the XML block attributes stored in the database and create object relations of the "XML linked" and "XML embedded" types if needed.

Step 4: Updating the system configuration

Updating INI settings for object relations

After running the upgrade scripts listed above you will need to configure how the object relations of the "XML linked" type will be handled on your site. From 3.9.0, inserting an internal link in the XML block attribute will lead to adding an "XML linked" relation. This behavior is controlled by the "ObjectRelationTyped" setting located in the "[BackwardCompatibilitySettings]" section of the "site.ini" configuration file. If you do not wish to use the "XML linked" type of relationships, make sure this setting is disabled.

Updating custom templates (optional)

In eZ Publish versions prior to 3.9, the "\$node" variable might be present in the pagelayout when view caching is disabled. From 3.9, the "\$node" variable is only available in node templates. This means that you will not be able to use this variable in the pagelayout (and other templates that are not related to the "view" view of the "content" module) even if view caching is disabled.

Enabling the Open Document Format extension (optional)

eZ Publish 3.9 includes the Open Document Format extension (previously known as "oo" and the "OpenOffice extension"). In order to use this extension, make sure your "settings/override/site.ini.append.php" file contains the following lines:

```
[ExtensionSettings]
ActiveExtensions []=ezodf
```

Note that it is also possible to enable this extension using the "Setup - Extensions" part of the administration interface as described here.

Updating INI settings for cronjobs (optional)

In eZ Publish 3.9.0, the default configuration of cronjobs has changed. The main set of cronjobs only contains tasks that should be run once a day. The tasks that should be executed frequently (workflow.php and notification.php) are now included in an additional set of cronjobs called "frequent". One more additional group of cronjobs called "infrequent" contains tasks that can be run once a week (linkcheck.php and basket_cleanup.php). The "ezpublish.cron" file is also updated. It is recommended to review your cronjob settings after upgrading.

Step 5: Clearing the caches

Whenever an eZ Publish solution is upgraded, all caches must be cleared in a proper way. This should be done from within a system shell:

1. Navigate into the eZ Publish 3.9.0 directory.
2. Run the clear cache script:

```
bin/shell/clearcache.sh --clear-all
```

Make sure that all caches are cleared. Sometimes the script is unable to clear caches because of restrictive file/directory permission settings. Make sure that all caches have been cleared by inspecting the contents of the various cache subdirectories within the "var" directory.

1.17 Upgrading to 3.8

This chapter contains information related to the upgrade of an eZ publish solution to [version 3.8](#). Please follow the upgrade procedures that are described in the following sections:

- Upgrading from 3.8.x to 3.8.y (page [280](#))
- Upgrading from 3.6.x or 3.7.x to 3.8.0

1.17.1 from 3.8.x to 3.8.y

This section describes how to upgrade your existing eZ Publish 3.8.x installation to version 3.8.y, for example from 3.8.0 to 3.8.10. If you are upgrading from a version prior to eZ Publish 3.8.0, you should first upgrade to 3.8.0 as described in this section.

Important note

Before continuing, note that version 3.8 is no longer supported. Instead of running eZ Publish 3.8, you should consider upgrading to one of the [supported versions](#). If you want to run 3.8 anyway, make sure you upgrade to version 3.8.10, which is the latest stable release in the 3.8 branch. Refer to the [changelogs](#) for more information about the issues that were fixed in the latest 3.8.x releases or view the short list of changes below.

- There is a significant problem with eZ Publish 3.8.0 - 3.8.8. More precisely, the discount functionality of the shop module contains a security vulnerability. The same bug exists in eZ Publish 3.9.0 - 3.9.2. The issue was fixed in eZ Publish 3.8.9, 3.9.3 and 3.10.0. Refer to the [announcement](#) for details.
- eZ Publish versions 3.8.0 - 3.8.9 contain a security vulnerability that can be used to show hidden content on a public siteaccess. The same bug exists in eZ Publish 3.9.0 - 3.9.3. This issue was fixed in eZ Publish 3.8.10, 3.9.4 and 3.10.0. Refer to the [announcement](#) for details.

Make sure that you have a working backup of the site before you do the actual upgrade. The upgrade procedure consists of the following steps:

1. Upgrading the distribution files to 3.8.10
2. Upgrading the database to 3.8.10
3. Running the system upgrade scripts
4. Updating the system configuration
5. Clearing the caches

Step 1: Upgrading the distribution files

The easiest way to upgrade the distribution files is to unpack eZ Publish 3.8.10 to a directory and then copy the directories that contain site-specific files from the existing installation. Make sure that you copy the following directories:

- design/example
- design/example_admin

- var
- settings/siteaccess
- settings/override

Replace "example" and "example_admin" with actual names used by your siteaccesses.

Important bug fix for the "user" module

In eZ Publish 3.8.10, the "register" view of the "user" module contains a bug. It prevents the system from registering new users when the "VerifyUserEmail" setting is enabled (it is enabled by default). The same bug exists in eZ Publish 3.9.4 and 3.10.0 (refer to <http://issues.ez.no/11598> for more information). This issue will be fixed in the upcoming eZ Publish 3.9.5 and 3.10.1 releases. In order to make your user registration mechanism work properly in 3.8.10, you have to download the updated version of "kernel/user/register.php" from <http://pubsvn.ez.no> and replace the corresponding file in your installation.

Custom extensions

If you are using custom extensions then the subdirectories inside the "extension" directory will also have to be copied. However, make sure that you do not overwrite any extensions that come with eZ Publish (for example the "PayPal" extension).

Step 2: Upgrading the database

To upgrade 3.8.0 database to 3.8.10, you should navigate into the eZ Publish 3.8.10 directory and run the following database upgrade scripts one after another:

1. dbupdate-3.8.0-to-3.8.1.sql
2. dbupdate-3.8.1-to-3.8.2.sql
3. dbupdate-3.8.2-to-3.8.3.sql
4. dbupdate-3.8.3-to-3.8.4.sql
5. dbupdate-3.8.4-to-3.8.5.sql
6. dbupdate-3.8.5-to-3.8.6.sql
7. dbupdate-3.8.6-to-3.8.7.sql
8. dbupdate-3.8.7-to-3.8.8.sql
9. dbupdate-3.8.8-to-3.8.9.sql
10. dbupdate-3.8.9-to-3.8.10.sql

Note that there are seven version-focused upgrade scripts, meaning that seven of the scripts in the list contain only two lines (that simply update the version number) and can be skipped. Therefore, the list of scripts to run contains three items instead of ten:

1. dbupdate-3.8.0-to-3.8.1.sql
2. dbupdate-3.8.4-to-3.8.5.sql
3. dbupdate-3.8.9-to-3.8.10.sql

Running these three scripts is equivalent to running all the ten scripts mentioned above.

MySQL

The database upgrade scripts are located in the "update/database/mysql/3.8/" directory of your eZ Publish installation. Each of these scripts can be launched using the following shell command:

```
mysql -u <username> -p<password> <database> < update/database/mysql/3.8/  
dbupdate-3.8.x-to-3.8.y.sql
```

PostgreSQL

The database upgrade scripts are located in the "update/database/postgresql/3.8/" directory of your eZ Publish installation. Each of these scripts can be launched using the following shell command:

```
psql -d <database> -U <downer> < update/database/postgresql/3.8/  
dbupdate-3.8.x-to-3.8.y.sql
```

Step 3: Running the system upgrade scripts

Custom classes in XML tags

In eZ Publish 3.8.0 and earlier versions, you could use custom classes in XML tags without having to specify them first. From eZ Publish 3.8.1, you have to specify the names of all custom classes that are used within XML tags. The names of the classes must be added to the global (or siteaccess) "content.ini.append.php" configuration file. For example, you will not be able to use "pRed" and "pBlue" classified paragraphs in your XML blocks unless you have the [paragraph] section in "content.ini.append.php" containing the following lines:

```
AvailableClasses []  
AvailableClasses []=pRed  
AvailableClasses []=pBlue
```

If you are using XML tag classes that are not listed in "content.ini" on your site, you will need to add the necessary configuration settings manually or run the "correctxmltextclasses.php" script. (If your installation does not include the "correctxmltextclasses.php" script, you need to download it [here](#).)

If all your siteaccesses are using the same database, run the script with the "--global" option as described below.

1. Navigate into the eZ Publish directory.
2. Run the script using the following shell command:

```
php update/common/scripts/correctxmltextclasses.php --global
```

The script will go through the XML blocks stored in the database, check which classes are in use and add missing settings (if any) to the "settings/override/content.ini.append.php" configuration file. If you do not wish to update this configuration file automatically, run the script like this:

```
php update/common/scripts/correctxmltextclasses.php --dump-only
```

The script will output the list of classes that need to be specified, for example:

```
Element 'paragraph': class 'pBlue' is not defined.  
Element 'paragraph': class 'pRed' is not defined.
```

In case if your siteaccesses are using different databases, the classes that are used within XML tags must be specified per siteaccess. This means that the upgrade script must be run for each of your siteaccesses.

1. Navigate into the eZ Publish directory.
2. Run the script (replace "example" with the actual name of your siteaccess):

```
php update/common/scripts/correctxmltextclasses.php -s example
```

The script will automatically update the "settings/siteaccess/example/content.ini.append.php" configuration file.

If the siteaccess is not specified then the default siteaccess will be used.

Changes to roles and policies

From 3.8.9, users (typically the anonymous role) need to have access to the "tipafriend" function of the "content" module in order to use the "Tip a friend" feature. Because of compatibility reasons, you need to run the "updatetipafriendpolicy.php" script.

Note for Windows users: in eZ Publish 3.8 the "updatetipafriendpolicy.php" script contains a bug that prevents it from granting access to the "Tip a friend" feature to the users. Refer to <http://issues.ez.no/11663> for more information about the problem and the solution.

The following example shows how to run the script:

1. Navigate into the eZ Publish 3.8.10 directory.
2. Run the script using the following shell command:

```
php update/common/scripts/3.8/  
updatetipafriendpolicy.php -s example_admin -l login -p password
```

Replace "example_admin" with the actual name of your admin siteaccess, use the administrator's login and password instead of "login" and "password".

The script will add a new role that grants access to the "Tip a friend" feature and ask for confirmation about assigning this role to each user or user group located one level beneath the "Users" top level node.

Step 4: Updating the system configuration

Updating INI settings for "html" classification

In eZ Publish versions from 3.8.0 to 3.8.3 it is possible to include HTML code in XML blocks by using the "html" classification. However, this feature can also be used to insert JavaScript code, making it possible for users with sufficient privileges to make an XSS exploit. This feature is therefore disabled by default in eZ Publish 3.8.4 and later, and should only be enabled if you really trust your editors.

To enable html classification, add the following lines to your "content.ini.append.php" file:

```
[literal]  
# The class 'html' is disabled by default because it gives editors the  
# possibility to insert html and javascript code in XML blocks.  
# Don't enable the 'html' class unless you really trust all users who has  
# privileges to edit objects containing XML blocks.  
AvailableClasses[]=html
```

Enabling the Online Editor extension (optional)

eZ Publish 3.8.1 and later versions include the Online Editor extension licensed under the [GPL](#). If you wish to use this extension, make sure it is enabled (refer to this section for more information).

Default configuration of unique user emails

In eZ Publish versions prior to 3.8.0, the default configuration does not allow different users to be registered with the exact same email address. This is just a built-in precaution mechanism which can be easily turned off by setting the "RequireUniqueEmail" directive within the [UserSettings] block of a configuration override for "site.ini" to "false". However, this default configuration has been accidentally changed in eZ Publish version 3.8.0 (refer to <http://issues.ez.no/9643> for more information). The change is reverted in eZ Publish version 3.8.7. If you are going to use eZ Publish 3.8.0-3.8.6, it is recommended to add the following line to the [UserSettings] block of the "override/site.ini.append.php" configuration file:

```
RequireUniqueEmail=true
```

Step 5: Clearing the caches

Whenever an eZ Publish solution is upgraded, all caches must be cleared in a proper way. This should be done from within a system shell:

1. Navigate into the eZ Publish 3.8.10 directory.
2. Run the clear cache script:

```
bin/shell/clearcache.sh --clear-all
```

Make sure that all caches are cleared. Sometimes the script is unable to clear caches because of restrictive file/directory permission settings. Make sure that all caches have been cleared by inspecting the contents of the various cache subdirectories within the "var" directory.

1.18 Upgrading to 3.7

This chapter contains information related to the upgrade of an eZ Publish solution to [version 3.7](#). Please follow the upgrade procedures that are described in the following sections:

- [Upgrading from 3.7.x to 3.7.y \(page 287\)](#)
- [Upgrading from 3.5.2 to 3.7.0 \(page 290\)](#)

1.18.1 from 3.7.x to 3.7.y

This section describes how to upgrade your existing eZ Publish 3.7.x installation to version 3.7.y, for example from 3.7.0 to 3.7.9. If you are upgrading from a version prior to eZ Publish 3.7.0, you should first upgrade to 3.7.0 as described in this section (page 290).

Please make sure that you have a working backup of the site before you do the actual upgrade. The upgrade procedure consists of the following steps:

1. Upgrading the distribution files to 3.7.9
2. Upgrading the database to 3.7.9
3. Running the system upgrade scripts
4. Updating INI settings for "html" classification
5. Clearing the caches

Step 1: Upgrading the distribution files

The easiest way to upgrade the distribution files is to unpack eZ Publish 3.7.9 to a directory and then copy the directories that contain site-specific files from the existing installation. Make sure that you copy the following directories:

- design/example
- design/example_admin
- var
- settings/siteaccess
- settings/override

Replace "example" and "example_admin" with actual names used by your siteaccesses.

Custom extensions

If you are using custom extensions then the sub-directories inside the "extension" directory will also have to be copied. However, make sure that you do not overwrite any extensions that come with eZ Publish (for example the "PayPal" extension).

Step 2: Upgrading the database

To upgrade 3.7.0 database to 3.7.9, you should navigate into the eZ Publish 3.7.9 directory and run the following database upgrade scripts one after another:

1. dbupdate-3.7.0-to-3.7.1.sql
2. dbupdate-3.7.1-to-3.7.2.sql
3. dbupdate-3.7.2-to-3.7.3.sql
4. dbupdate-3.7.3-to-3.7.4.sql
5. dbupdate-3.7.4-to-3.7.5.sql
6. dbupdate-3.7.5-to-3.7.6.sql
7. dbupdate-3.7.6-to-3.7.7.sql
8. dbupdate-3.7.7-to-3.7.8.sql
9. dbupdate-3.7.8-to-3.7.9.sql

MySQL

The database upgrade scripts are located in the "update/database/mysql/3.7/" directory of your eZ Publish installation. Each of these scripts can be launched using the following shell command:

```
mysql -u <username> -p<password> <database> < update/database/mysql/3.7/  
dbupdate-3.7.x-to-3.7.y.sql
```

PostgreSQL

The database upgrade scripts are located in the "update/database/postgresql/3.7/" directory of your eZ Publish installation. Each of these scripts can be launched using the following shell command:

```
psql -d <database> -U <dbowner> < update/database/postgresql/3.7/  
dbupdate-3.7.x-to-3.7.y.sql
```

Step 3: Running the system upgrade scripts

There are currently no upgrade scripts for upgrading from 3.7.x to 3.7.y.

Step 4: Updating INI settings for "html" classification

In eZ Publish versions from 3.7.3 to 3.7.8 it is possible to include HTML code in XML blocks by using the "html" classification. However, this feature can also be used to insert JavaScript code, making it possible for users with sufficient privileges to make an XSS exploit. This feature is therefore disabled by default in eZ Publish 3.7.9 and later, and should only be enabled if you really trust your editors.

To enable html classification, add the following lines to your "content.ini.append.php" file:

```
[literal]
# The class 'html' is disabled by default because it gives editors the
# possibility to insert html and javascript code in XML blocks.
# Don't enable the 'html' class unless you really trust all users who has
# privileges to edit objects containing XML blocks.
AvailableClasses[]=html
```

Step 5: Clearing the caches

Whenever an eZ Publish solution is upgraded, all caches must be cleared in a proper way. This should be done from within a system shell:

1. Navigate into the eZ Publish 3.7.9 directory.
2. Run the clear cache script:

```
bin/shell/clearcache.sh --clear-all
```

Please make sure that all caches are cleared. Sometimes the script is unable to clear caches because of restrictive file/directory permission settings. Make sure that all caches have been cleared by inspecting the contents of the various cache sub-directories within the "var" directory.

1.18.2 from 3.5.2 to 3.7.0

This section describes how to upgrade your existing eZ Publish 3.5.2 installation to version 3.7.0. If you are upgrading from a version prior to eZ Publish 3.5.2, you need to first upgrade to 3.5.2 (page 307) before you can upgrade to 3.7.0.

Make sure that you have a working backup of the site before you do the actual upgrade. The upgrade procedure consists of the following steps:

1. Upgrading the distribution files to 3.7.0
2. Upgrading the database to 3.7.0
3. Running the 3.6.0 upgrade scripts
4. Updating the system configuration
5. Clearing the caches

Step 1: Upgrading the distribution files

The easiest way to upgrade the distribution files is to unpack eZ Publish 3.7.0 to a directory and then copy the directories that contain site-specific files from the existing installation. Make sure that you copy the following directories:

- design/example
- design/example_admin
- var
- settings/siteaccess
- settings/override

Replace "example" and "example_admin" with actual names used by your siteaccesses.

Custom extensions

If you are using custom extensions then the subdirectories inside the "extension" directory will also have to be copied. However, make sure that you do not overwrite any extensions that come with eZ Publish (for example the "PayPal" extension).

Step 2: Upgrading the database

The following text describes how a 3.5.2 database can be upgraded to 3.7.0.

MySQL

1. Navigate into the eZ Publish 3.7.0 directory.
2. Run the first database upgrade script:

```
mysql -u <username> -p<password> <database> < update/database/mysql/3.6/  
dbupdate-3.5.2-to-3.6.0.sql
```

3. Run the second database upgrade script:

```
mysql -u <username> -p<password> <database> < update/database/mysql/3.7/  
dbupdate-3.6.0-to-3.7.0.sql
```

Note that the CREATE TABLE statements in the database upgrade scripts do not specify which storage engine to use (no ENGINE or TYPE option), and thus the default storage engine will be used. Normally, it is MyISAM (starting from MySQL v.3.23). If you are using InnoDB, make sure the default storage engine is set to InnoDB before you run the database upgrade script (refer to [MySQL documentation](#) for information about how to set the default engine). If you were not able to change the MySQL configuration on your server, and the upgrade left you with a mix of table types, you can use the "bin/php/ezconvertmysqltabletype.php" script for database conversion. It is also possible to convert the newly created tables to InnoDB using ALTER TABLE statements as shown in the following example:

```
ALTER TABLE table_name1 TYPE = innodb;  
ALTER TABLE table_name2 TYPE = innodb;  
...
```

Replace "table_name1", "table_name2" with the actual names of the tables that need to be converted.

PostgreSQL

1. Navigate into the eZ Publish 3.7.0 directory.
2. Run the first database upgrade script:

```
psql -d <database> -U <downer> < update/database/postgresql/3.6/  
dbupdate-3.5.2-to-3.6.0.sql
```

3. Run the second database upgrade script:

```
psql -d <database> -U <downer> < update/database/postgresql/3.7/  
dbupdate-3.6.0-to-3.7.0.sql
```

Step 3: Running the 3.6.0 upgrade scripts

The 3.6.0 version of eZ Publish introduced a couple of new features that are used in 3.7 version also. In order to make sure that your site is compatible with these features, you'll have to run a couple of upgrade scripts.

Internal linking

In order to be compatible with the [new internal linking feature](#), you will have to run the "convertxmllinks.php" script. This script should be run for all siteaccesses that use different databases. If you only have a public and an administration siteaccess (which is the most typical/usual case), then you will only need to run the script for one of the siteaccesses.

1. Navigate into the eZ Publish 3.7.0 directory.
2. Run the script (replace <siteaccess> with the name of your siteaccess(es)):

```
php update/common/scripts/convertxmllinks.php -s <siteaccess>
```

Advanced related objects

In order to be compatible with the [new advanced related objects feature](#), you will have to run the "updaterelatedobjectlinks.php" script. This script should be run for all siteaccesses that use different databases. If you only have a public and an administration siteaccess (which is the most typical/usual case), then you will only need to run the script for one of the siteaccesses.

1. Navigate into the eZ Publish 3.7.0 directory.
2. Run the script (replace <siteaccess> with the name of your siteaccess(es)):

```
php update/common/scripts/updaterelatedobjectlinks.php -s <siteaccess>
```

Step 4: Updating the system configuration

Administration interface toolbar changes

The right section of the administration interface has been slightly changed in 3.6.0. It now uses the toolbar system and features a couple of new toolbars for developers (clear cache, debug control, etc.). In order to make it work, you will have to add the following settings in "toolbar.ini.append.php" for the siteaccesses that use the admin design:

```

[Toolbar]
AvailableToolBarArray[]=admin_right
AvailableToolBarArray[]=admin_developer

[Tool]
AvailableToolArray[]=admin_current_user
AvailableToolArray[]=admin_bookmarks
AvailableToolArray[]=admin_clear_cache
AvailableToolArray[]=admin_quick_settings

[Toolbar_admin_right]
Tool[]
Tool[]=admin_current_user
Tool[]=admin_bookmarks

[Toolbar_admin_developer]
Tool[]
Tool[]=admin_clear_cache
Tool[]=admin_quick_settings

```

Visibility of hidden nodes

In eZ Publish 3.6.0, the default value of the "ShowHiddenNodes" configuration directive has become "false". This means that hidden nodes will not be available on any siteaccess. It is recommended to add the following line to the "[SiteAccessSettings]" section of the "settings/siteaccess/example_admin/site.ini.append.php" configuration file (replace "example_admin" by the actual name of your admin siteaccess):

```
ShowHiddenNodes=true
```

This will instruct the system to show hidden nodes in the administration interface.

Enabling database transaction support (optional)

eZ Publish 3.6.0 introduced support for database transactions. This feature makes eZ Publish less vulnerable for database errors and inconsistencies due to aborted requests. It is highly recommended to make use of this feature. Transaction support is available for MySQL from version 4.1 and above and for all PostgreSQL and Oracle versions. MySQL users must first convert the database (see below).

MySQL table conversion

MySQL users have to perform an extra step when enabling transaction support. Make sure that you have MySQL 4.0 or later before attempting to perform this step. Previous versions of eZ

Publish use "MyISAM" type tables. Such tables do not support transactions. The database must be converted so that all tables use the "InnoDB" type instead of "MyISAM". This can be done using the "ezconvertmysqltabletype.php" script:

1. Navigate into the eZ Publish 3.7.0 directory.
2. Run the database conversion script; example:

```
bin/php/ezconvertmysqltabletype.php --host=localhost --user=arnold  
--password=secret --database=ezpublish --newtype=innodb
```

(You will have to provide your own values for the "host", "user", "password" and "database" parameters.)

It is also recommended to set the default storage engine to InnoDB (refer to [MySQL documentation](#) for information about how this can be done).

Enabling transaction support in eZ Publish

Each siteaccess which is configured to communicate with a database that uses "InnoDB" tables can be configured to use transactions. This can be done by adding "Transactions=enabled" within the "[DatabaseSettings]" block in "site.ini.append.php" for the target siteaccess(es). You can also add this line to the "settings/override/site.ini.append.php" file instead of doing it for each of the target siteaccesses (this way is recommended if all your siteaccesses use the same database).

Step 5: Clearing the caches

Whenever an eZ Publish solution is upgraded, all caches must be cleared in a proper way. This should be done from within a system shell:

1. Navigate into the eZ Publish 3.7.0 directory.
2. Run the clear cache script:

```
bin/shell/clearcache.sh --clear-all
```

Make sure that all caches are cleared. Sometimes the script is unable to clear caches because of restrictive file/directory permission settings. Make sure that all caches have been cleared by inspecting the contents of the various cache subdirectories within the "var" directory.

1.19 Upgrading to 3.6

This chapter contains information related to the upgrade of an eZ Publish solution to version 3.6. Please follow the upgrade procedures that are described in the following sections:

- Upgrading from 3.6.x to 3.6.y (page [296](#))
- Upgrading from 3.5.2 to 3.6.0 (page [299](#))

1.19.1 from 3.6.x to 3.6.y

This section describes how to upgrade your existing eZ publish 3.6.x installation to version 3.6.y, for example from 3.6.0 to 3.6.11. If you are upgrading from a version prior to eZ publish 3.6.0, you should first upgrade to 3.6.0 as described in this section (page 299).

Please make sure that you have a working backup of the site before you do the actual upgrade. The upgrade procedure consists of the following steps:

1. Upgrading the distribution files to 3.6.11
2. Upgrading the database to 3.6.11
3. Running the system upgrade scripts
4. Updating INI settings for "html" classification
5. Clearing the caches

Step 1: Upgrading the distribution files

The easiest way to upgrade the distribution files is to unpack eZ publish 3.6.11 to a directory and then copy the directories that contain site-specific files from the existing installation. Make sure that you copy the following directories:

- design/example
- design/example_admin
- var
- settings/siteaccess
- settings/override

Replace "example" and "example_admin" with actual names used by your siteaccesses.

Custom extensions

If you are using custom extensions then the subdirectories inside the "extension" directory will also have to be copied. However, make sure that you do not overwrite any extensions that come with eZ publish (for example the "PayPal" extension).

Step 2: Upgrading the database

To upgrade 3.6.0 database to 3.6.11, you should navigate into the eZ publish 3.6.11 directory and run the following database upgrade scripts one after another:

1. dbupdate-3.6.0-to-3.6.1.sql
2. dbupdate-3.6.1-to-3.6.2.sql
3. dbupdate-3.6.2-to-3.6.3.sql
4. dbupdate-3.6.3-to-3.6.4.sql
5. dbupdate-3.6.4-to-3.6.5.sql
6. dbupdate-3.6.5-to-3.6.6.sql
7. dbupdate-3.6.6-to-3.6.7.sql
8. dbupdate-3.6.7-to-3.6.8.sql
9. dbupdate-3.6.8-to-3.6.9.sql
10. dbupdate-3.6.9-to-3.6.10.sql
11. dbupdate-3.6.10-to-3.6.11.sql

MySQL

The database upgrade scripts are located in the "update/database/mysql/3.6/" directory of your eZ publish installation. Each of these scripts can be launched using the following shell command:

```
mysql -u <username> -p<password> <database> < update/database/mysql/3.6/  
dbupdate-3.6.x-to-3.6.y.sql
```

PostgreSQL

The database upgrade scripts are located in the "update/database/postgresql/3.6/" directory of your eZ publish installation. Each of these scripts can be launched using the following shell command:

```
psql -d <database> -U <downer> < update/database/postgresql/3.6/  
dbupdate-3.6.x-to-3.6.y.sql
```

Step 3: Running the system upgrade scripts

There are currently no upgrade scripts for upgrading from 3.6.x to 3.6.y.

Step 4: Updating INI settings for "html" classification

In eZ publish versions from 3.6.5 to 3.6.10 it is possible to include HTML code in XML blocks by using the "html" classification. However, this feature can also be used to insert JavaScript code, making it possible for users with sufficient privileges to make an XSS exploit. This feature is therefore disabled by default in eZ publish 3.6.11 and later, and should only be enabled if you really trust your editors.

To enable html classification, add the following lines to your "content.ini.append.php" file:

```
[literal]
# The class 'html' is disabled by default because it gives editors the
# possibility to insert html and javascript code in XML blocks.
# Don't enable the 'html' class unless you really trust all users who has
# privileges to edit objects containing XML blocks.
AvailableClasses[]=html
```

Step 5: Clearing the caches

Whenever an eZ publish solution is upgraded, all caches must be cleared in a proper way. This should be done from within a system shell:

1. Navigate into the eZ publish 3.6.11 directory.
2. Run the clear cache script:

```
bin/shell/clearcache.sh --clear-all
```

Please make sure that all caches are cleared. Sometimes the script is unable to clear caches because of restrictive file/directory permission settings. Make sure that all caches have been cleared by inspecting the contents of the various cache subdirectories within the "var" directory.

1.19.2 from 3.5.2 to 3.6.0

This section describes how to upgrade your existing eZ Publish 3.5.2 installation to version 3.6.0. If you are upgrading from a version prior to eZ Publish 3.5.2, you need to first upgrade to 3.5.2 (page 307) before you can upgrade to 3.6.0.

Make sure that you have a working backup of the site before you do the actual upgrade. The upgrade procedure consists of the following steps:

1. Upgrading the distribution files to 3.6.0
2. Upgrading the database to 3.6.0
3. Running the 3.6.0 upgrade scripts
4. Updating the system configuration
5. Clearing the caches

Step 1: Upgrading the distribution files

The easiest way to upgrade the distribution files is to unpack eZ Publish 3.6.0 to a directory and then copy the directories that contain site-specific files from the existing installation. Make sure that you copy the following directories:

- design/example
- design/example_admin
- var
- settings/siteaccess
- settings/override

Replace "example" and "example_admin" with actual names used by your siteaccesses.

Custom extensions

If you are using custom extensions then the subdirectories inside the "extension" directory will also have to be copied. However, make sure that you do not overwrite any extensions that come with eZ Publish (for example the "PayPal" extension).

Step 2: Upgrading the database

The following text describes how a 3.5.2 database can be upgraded to 3.6.0.

MySQL

1. Navigate into the eZ Publish 3.6.0 directory.
2. Run the database upgrade script:

```
mysql -u <username> -p<password> <database> < update/database/mysql/3.6/  
dbupdate-3.5.2-to-3.6.0.sql
```

Note that the CREATE TABLE statements in the database upgrade script do not specify which storage engine to use (no ENGINE or TYPE option), and thus the default storage engine will be used. Normally, it is MyISAM (starting from MySQL v.3.23). If you are using InnoDB, make sure the default storage engine is set to InnoDB before you run the database upgrade script (refer to [MySQL documentation](#) for information about how to set the default engine). If you were not able to change the MySQL configuration on your server, and the upgrade left you with a mix of table types, you can use the "bin/php/ezconvertmysqltabletype.php" script for database conversion. It is also possible to convert the newly created tables to InnoDB using ALTER TABLE statements as shown in the following example:

```
ALTER TABLE table_name1 TYPE = innodb;  
ALTER TABLE table_name2 TYPE = innodb;  
...
```

Replace "table_name1", "table_name2" with the actual names of the tables that need to be converted.

PostgreSQL

1. Navigate into the eZ Publish 3.6.0 directory.
2. Run the database upgrade script:

```
psql -d <database> -U <dbowner> < update/database/postgresql/3.6/  
dbupdate-3.5.2-to-3.6.0.sql
```

Step 3: Running the 3.6.0 upgrade scripts

The 3.6.0 version of eZ Publish introduces a couple of new features. In order to make sure that your site is compatible with these feature, you'll have to run a couple of upgrade scripts.

Internal linking

In order to be compatible with the [new internal linking feature](#), you will have to run the "convertxmllinks.php" script. This script should be run for all siteaccesses that use different databases.

If you only have a public and an administration siteaccess (which is the most typical/usual case), then you will only need to run the script for one of the siteaccesses. If you do not specify any siteaccess when running the script, then the default siteaccess will be used.

1. Navigate into the eZ Publish 3.6.0 directory.
2. Run the script for your default siteaccess:

```
php update/common/scripts/convertxmllinks.php
```

or use the following command to run the script for a specific siteaccess (replace <siteaccess> with the name of your siteaccess):

```
php update/common/scripts/convertxmllinks.php -s <siteaccess>
```

Advanced related objects

In order to be compatible with the [new advanced related objects feature](#), you will have to run the "updaterelatedobjectlinks.php" script. This script should be run for all siteaccesses that use different databases. If you only have a public and an administration siteaccess (which is the most typical/usual case), then you will only need to run the script for one of the siteaccesses.

1. Navigate into the eZ Publish 3.6.0 directory.
2. Run the script (replace <siteaccess> with the name of your siteaccess(es)):

```
php update/common/scripts/updaterelatedobjectlinks.php -s <siteaccess>
```

Converting the "eztime" attributes

In eZ Publish 3.5.3 and 3.6.0, the "Time" datatype has been modified in order to make it possible to interpret the value of hours and minutes properly and independently from DST and GMT transformations (see the [feature doc](#) for more information). When upgrading from 3.5.2, you need to run the "updateeztimetype.php" script in order to convert the attributes of the "eztime" datatype from GMT to the server local time (replace "example" with the name of the siteaccess):

```
php update/common/scripts/updateeztimetype.php -s example
```

It is recommended to create a database backup before using this script. The script must be run for each siteaccess. Note that there is no need to run this script when upgrading from 3.5.3 or later.

Step 4: Updating the system configuration

Administration interface toolbar changes

The right section of the administration interface has been slightly changed in 3.6.0. It now uses the toolbar system and features a couple of new toolbars for developers (clear cache, debug control, etc.). In order to make it work, you will have to add the following settings in "toolbar.ini.append.php" for the siteaccesses that use the admin design:

```
[Toolbar]
AvailableToolBarArray[]=admin_right
AvailableToolBarArray[]=admin_developer

[Tool]
AvailableToolArray[]=admin_current_user
AvailableToolArray[]=admin_bookmarks
AvailableToolArray[]=admin_clear_cache
AvailableToolArray[]=admin_quick_settings

[Toolbar_admin_right]
Tool[]
Tool[]=admin_current_user
Tool[]=admin_bookmarks

[Toolbar_admin_developer]
Tool[]
Tool[]=admin_clear_cache
Tool[]=admin_quick_settings
```

Visibility of hidden nodes

In eZ Publish 3.6.0, the default value of the "ShowHiddenNodes" configuration directive has become "false". This means that hidden nodes will not be available on any siteaccess. It is recommended to add the following line to the "[SiteAccessSettings]" section of the "settings/siteaccess/example_admin/site.ini.append.php" configuration file (replace "example_admin" by the actual name of your admin siteaccess):

```
ShowHiddenNodes=true
```

This will instruct the system to show hidden nodes in the administration interface.

XML tag changes

From 3.6, the "object" XML tag is deprecated (but not removed because of backwards compatibility) and you should use the new "embed" tag for object embedding. The Online Editor auto-

matically converts "object" tags to "embed" but this may cause problems with custom classes. To fix this, check whether your "content.ini.append.php" configuration file contains custom classes for the "object" tag. If yes, specify the same custom classes for the "embed" tag.

Note that if you use alternate/custom templates for rendering the content of XML tags, you will probably need to add one more template for "embed".

Example 1

Let's say that you have the following lines in an override of the "settings/content.ini" configuration file:

```
[object]
AvailableClasses []=imageRed
AvailableClasses []=imageBlue
```

If you are going to use the new "embed" tag, then you should add the following lines to the same file:

```
[embed]
AvailableClasses []=imageRed
AvailableClasses []=imageBlue
```

Example 2

If you have a custom template called "object.tpl" inside the "templates/content/datatype/view/ezxmltags/" subdirectory of your design, then you should add a new template called "embed.tpl" to the same directory.

Example 3

Let's say that your siteaccess "override.ini.append.php" file contains the following lines:

```
[imageRed_object]
Source=content/datatype/view/ezxmltags/object.tpl
MatchFile=imageRed_object.tpl
Subdir=templates
Match[classification]=imageRed
```

This means that all images classified as "imageRed" are rendered using the "imageRed_object.tpl" template located inside the "override/templates/" directory of your design. Since images will be inserted using the "embed" tag, you will have to override the "content/datatype/view/ezxmltags/embed.tpl" template in the same way.

Enabling database transaction support (optional)

eZ Publish 3.6.0 introduces support for database transactions. This feature makes eZ Publish less vulnerable for database errors and inconsistencies due to aborted requests. It is highly recommended to make use of this feature. Transaction support is available for MySQL from version 4.1 and above and for all PostgreSQL and Oracle versions. MySQL users must first convert the database (see below).

MySQL table conversion

MySQL users have to perform an extra step when enabling transaction support. Make sure that you have MySQL 4.0 or later before attempting to perform this step. Previous versions of eZ Publish use "MyISAM" type tables. Such tables do not support transactions. The database must be converted so that all tables use the "InnoDB" type instead of "MyISAM". This can be done using the "ezconvertmysqltabletype.php" script:

1. Navigate into the eZ Publish 3.6.0 directory.
2. Run the database conversion script; example:

```
bin/php/ezconvertmysqltabletype.php --host=localhost --user=arnold  
--password=secret --database=ezpublish --newtype=innodb
```

(You will have to provide your own values for the "host", "user", "password" and "database" parameters.)

It is also recommended to set the default storage engine to InnoDB (refer to [MySQL documentation](#) for information about how this can be done).

Enabling transaction support in eZ Publish

Each siteaccess which is configured to communicate with a database that uses "InnoDB" tables can be configured to use transactions. This can be done by adding "Transactions=enabled" within the "[DatabaseSettings]" block in "site.ini.append.php" for the target siteaccess(es). You can also add this line to the "settings/override/site.ini.append.php" file instead of doing it for each of the target siteaccesses (this way is recommended if all your siteaccesses use the same database).

Step 5: Clearing the caches

Whenever an eZ Publish solution is upgraded, all caches must be cleared in a proper way. This should be done from within a system shell:

1. Navigate into the eZ Publish 3.6.0 directory.

2. Run the clear cache script:

```
bin/shell/clearcache.sh --clear-all
```

Make sure that all caches are cleared. Sometimes the script is unable to clear caches because of restrictive file/directory permission settings. Make sure that all caches have been cleared by inspecting the contents of the various cache subdirectories within the "var" directory.

1.20 Upgrading to 3.5

This chapter contains information related to the upgrade of an eZ publish solution to version 3.5. Please follow the upgrade procedures that are described in the following sections:

- Upgrading from 3.5.x to 3.5.y (page [307](#))
- Upgrading from 3.4.4 to 3.5.0 (page [311](#))

1.20.1 from 3.5.x to 3.5.y

This section describes how to upgrade your existing eZ publish 3.5.x installation to version 3.5.y, for example from 3.5.0 to 3.5.11. If you are upgrading from a version prior to eZ publish 3.5.0, you should first upgrade to 3.5.0 as described here (page [311](#)).

Please make sure that you have a working backup of the site before you do the actual upgrade. The upgrade procedure consists of the following steps:

1. Upgrading the distribution files to 3.5.11
2. Upgrading the database to 3.5.11
3. Running the system upgrade scripts
4. Updating the system configuration
5. Clearing the caches

Step 1: Upgrading the distribution files

The easiest way to upgrade the distribution files is to unpack eZ publish 3.5.11 to a directory and then copy the directories that contain site-specific files from the existing installation. Make sure that you copy the following directories:

- design/example
- design/example_admin
- var
- settings/siteaccess
- settings/override

Replace "example" and "example_admin" with actual names used by your siteaccesses.

Custom extensions

If you are using custom extensions then the subdirectories inside the "extension" directory will also have to be copied. However, make sure that you do not overwrite any extensions that come with eZ publish (for example the "PayPal" extension).

Step 2: Upgrading the database

To upgrade 3.5.0 database to 3.5.11, you should navigate into the eZ publish 3.5.11 directory and run the following database upgrade scripts one after another:

1. dbupdate-3.5.0-to-3.5.1.sql
2. dbupdate-3.5.1-to-3.5.2.sql
3. dbupdate-3.5.2-to-3.5.3.sql
4. dbupdate-3.5.3-to-3.5.4.sql
5. dbupdate-3.5.4-to-3.5.5.sql
6. dbupdate-3.5.5-to-3.5.6.sql
7. dbupdate-3.5.6-to-3.5.7.sql
8. dbupdate-3.5.7-to-3.5.8.sql
9. dbupdate-3.5.8-to-3.5.9.sql
10. dbupdate-3.5.9-to-3.5.10.sql
11. dbupdate-3.5.10-to-3.5.11.sql

MySQL

The database upgrade scripts are located in the "update/database/mysql/3.5/" directory of your eZ publish installation. Each of these scripts can be launched using the following shell command:

```
mysql -u <username> -p<password> <database> < update/database/mysql/3.5/  
dbupdate-3.5.x-to-3.5.y.sql
```

PostgreSQL

The database upgrade scripts are located in the "update/database/postgresql/3.5/" directory of your eZ publish installation. Each of these scripts can be launched using the following shell command:

```
psql -d <database> -U <dbowner> < update/database/postgresql/3.5/  
dbupdate-3.5.x-to-3.5.y.sql
```

Step 3: Running the system upgrade scripts

Top level nodes/objects changes

In eZ publish 3.5.1, a new top level node/object called "Design" has been added. You need to run the "updatetoplevel.php" script for your admin siteaccess in order to make sure that all your top level nodes are properly configured (replace "example_admin" with the actual name of your admin siteaccess):

```
./update/common/scripts/updatetoplevel.php -s example_admin
```

The script will check the top level nodes and match them with the values defined in the "menu.ini" configuration file or its override. If "menu.ini" or its override contains incorrect information about the top level nodes, it will be automatically updated. If some of the top level nodes are missing, these will be created. (Please note that this script has been improved in eZ publish 3.5.2. If you are upgrading to a version prior to 3.5.2, it is recommended to copy the script from 3.5.2.)

Converting the "eztime" attributes

In eZ publish 3.5.3, the "Time" datatype has been modified in order to make it possible to interpret the value of hours and minutes properly and independently from DST and GMT transformations (see the [feature doc](#) for more information). You need to run the "updateeztimetype.php" script in order to convert the attributes of the "eztime" datatype from GMT to the server local time (replace "example" with the name of the siteaccess):

```
./update/common/scripts/updateeztimetype.php -s example
```

Please note that it is recommended to create a database backup before using this script. The script must be run for each siteaccess.

Changes related to 32bit/64bit compatibility

Since some PHP functions (like [crc32](#) and [ip2long](#)) work in different ways on 64bit and 32bit servers, changes have been made in eZ publish 3.5.8 in order to avoid problems when migrating from 32bit to 64bit architecture. You need to run the "updatecrc32.php" script when upgrading even if you don't have a 64bit machine:

```
./update/common/scripts/updatecrc32.php
```

The script will update old signed to unsigned values in the database. It is also recommended to specify "64bitCompatibilityMode=enabled" in your "site.ini.append.php" configuration file.

Step 4: Updating the system configuration

You are not required to do any configuration changes when upgrading from 3.5.x to 3.5.y. However, it is recommended to specify "64bitCompatibilityMode=enabled" in your "site.ini.append.php" configuration file after running the "updatecrc32.php" script.

Step 5: Clearing the caches

Whenever an eZ publish solution is upgraded, all caches must be cleared in a proper way. This should be done from within a system shell:

1. Navigate into the eZ publish 3.5.11 directory.
2. Run the clear cache script:

```
bin/shell/clearcache.sh --clear-all
```

Please make sure that all caches are cleared. Sometimes the script is unable to clear caches because of restrictive file/directory permission settings. Make sure that all caches have been cleared by inspecting the contents of the various cache subdirectories within the "var" directory.

1.20.2 from 3.4.4 to 3.5.0

This section describes how to upgrade your existing eZ publish 3.4.4 installation to version 3.5.0. If you are upgrading from a version prior to eZ publish 3.4.4, you need to first upgrade to 3.4.4 (page 317) before you can upgrade to 3.5.0.

Please make sure that you have a working backup of the site before you do the actual upgrade. The upgrade procedure consists of the following steps:

1. Upgrading the distribution files to 3.5.0
2. Upgrading the database to 3.5.0
3. Running the 3.5.0 upgrade scripts
4. Updating the system configuration
5. Clearing the caches

Step 1: Upgrading the distribution files

The easiest way to upgrade the distribution files is to unpack eZ publish 3.5.0 to a directory and then copy the directories that contain site-specific files from the existing installation. Make sure that you copy the following directories:

- design/example
- design/example_admin
- var
- settings/siteaccess
- settings/override

Replace "example" and "example_admin" with actual names used by your siteaccesses.

Custom extensions

If you are using custom extensions then the subdirectories inside the "extension" directory will also have to be copied. However, make sure that you do not overwrite any extensions that come with eZ publish (for example the "PayPal" extension).

Step 2: Upgrading the database

The following text describes how a 3.4.4 database can be upgraded to 3.5.0.

MySQL

1. Navigate into the eZ publish 3.5.0 directory.
2. Run the database upgrade script:

```
mysql -u <username> -p<password> <database> < update/database/mysql/3.5/  
dbupdate-3.4.4-to-3.5.0.sql
```

PostgreSQL

1. Navigate into the eZ publish 3.5.0 directory.
2. Run the database upgrade script:

```
psql -d <database> -U <downer> < update/database/postgresql/3.5/  
dbupdate-3.4.4-to-3.5.0.sql
```

Step 3: Running the 3.5.0 upgrade scripts

The 3.5.0 version of eZ publish introduces a couple of new features. In order to make sure that your site is compatible with these features, you'll have to run a couple of upgrade scripts.

Content class structure changes

In eZ publish 3.5.0, a new attribute called "is_container" has been added to content classes (see the [feature doc](#) for more information). The container flag controls whether an instance of the class should be allowed to have sub items (often called child nodes, children) or not. In order to be compatible with this functionality, you will have to run the "updateiscontainer.php" script for all siteaccesses that use different databases. If you only have a public and an administration siteaccess (which is the most typical/usual case), then you will only need to run the script for one of the siteaccesses:

1. Navigate into the eZ publish 3.5.0 directory.
2. Run the script (replace "example" with the name of your siteaccess(es)):

```
php update/common/scripts/updateiscontainer.php -s example
```

Please note that the script can only modify the general purpose classes. You should manually set the container flag for your custom classes (if there are any), otherwise it won't be possible to create subitems.

Updating the nice URLs (optional)

eZ publish 3.5 includes the built-in transliteration feature that allows to automatically convert special characters to their equivalent in English letters when generating the virtual URLs (nice URLs, URL aliases). For example, the Norwegian characters ””, ”” and ”” are converted to ”ae”, ”oe” and ”aa” instead of underscores. It is recommended (but not required) to run the ”updateniceurls.php” script in order to update your nice URLs:

1. Navigate into the eZ publish 3.5.0 directory.
2. Run the script:

```
php update/common/scripts/updateniceurls.php
```

Updating the search index (optional)

eZ publish 3.5.0 includes a few bug fixes related to the search index generation (see the [3.5.0 changelog](#) for more information). It is recommended (but not required) to run the ”update-searchindex.php” script with the ”--clean” option in order to update the search index of your site:

1. Navigate into the eZ publish 3.5.0 directory.
2. Run the script:

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

The ”--clean” option (added in eZ publish 3.3-1) means that the existing search index data should be removed before starting indexing.

Step 4: Updating the system configuration

Administration interface changes

The administration interface has been completely renewed in 3.5.0, and it requires two new override files. These are created automatically for new installations but should be copied manually when upgrading:

```
cp settings/siteaccess/admin/override.ini.append.php settings/siteaccess/  
example_admin/
```

```
cp settings/siteaccess/admin/icon.ini.append settings/siteaccess/example_admin/
```

Replace "example_admin" with the actual name of your admin siteaccess.

In order to be compatible with new functionality, you will also need to specify the "CachedView-Preferences" setting in the "site.ini.append.php" configuration file of your admin siteaccess:

```
[ContentSettings]
CachedViewPreferences[full]=admin_navigation_content=0;
admin_navigation_details=0;admin_navigation_languages=0;
admin_navigation_locations=0;admin_navigation_relations=
0;admin_navigation_roles=0;admin_navigation_policies=0;
admin_navigation_content=0;admin_navigation_translations=0;
admin_children_viewmode=list;admin_list_limit=1;admin
_edit_show_locations=0;admin_url_list_limit=10;admin_url_view_limit=10;
admin_section_list_limit=1;admin_orderlist_sortfield=user_name;
admin_orderlist_sortorder=desc;ad
min_search_stats_limit=1;admin_treemenu=1;admin_bookmarkmenu=1;
admin_left_menu_width=13
```

If you have a large site that contains a lot of content objects, then it may be useful to reduce the number of objects shown in the left tree menu in the administration interface. It is possible to configure the tree menu limitations so that it for example only shows certain types of nodes, only goes to a certain depth and so on. Configuration should be done by creating an override for the "contentstructuremenu.ini" file.

Configuring the smart viewcache cleaning system (optional)

If you have not configured the [smart viewcache cleaning system](#) before (this feature is available from version 3.4.2), then you will probably need to do this. This feature allows to determine which additional nodes the system should clear the caches for when the content of an object is changed (by default, the view cache will be cleared for all published nodes of this object, their parent nodes, nodes of reverse related objects and nodes of objects with the same keyword). Please note that this step is only required if you are running a complex site or if you have cache problems after upgrading (for example, if your changes to content are not visible on some pages).

New folders under the "Media" top level node

You need to create three new folders called "Files", "Images" and "Multimedia" under the "Media" top level node. These will be used for autoplacement of uploaded files.

Step 5: Clearing the caches

Whenever an eZ publish solution is upgraded, all caches must be cleared in a proper way. This should be done from within a system shell:

1. Navigate into the eZ publish 3.5.0 directory.

2. Run the clear cache script:

```
bin/shell/clearcache.sh --clear-all
```

Please make sure that all caches are cleared. Sometimes the script is unable to clear caches because of restrictive file/directory permission settings. Make sure that all caches have been cleared by inspecting the contents of the various cache subdirectories within the "var" directory.

1.21 Upgrading to 3.4

This chapter contains information related to the upgrade of an eZ publish solution to [version 3.4](#). Please follow the upgrade procedures that are described in the following sections:

- [Upgrading from 3.4.x to 3.4.y \(page 317\)](#)
- [Upgrading from 3.3-5 to 3.4.0 \(page 320\)](#)

1.21.1 from 3.4.x to 3.4.y

This section describes how to upgrade your existing eZ publish 3.4.x installation to version 3.4.y, for example from 3.4.0 to 3.4.8. If you are upgrading from a version prior to eZ publish 3.4.0, you should first upgrade to 3.4.0 as described here (page [320](#)).

Please make sure that you have a working backup of the site before you do the actual upgrade. The upgrade procedure consists of the following steps:

1. Upgrading the distribution files to 3.4.8
2. Upgrading the database to 3.4.8
3. Running the system upgrade scripts
4. Configuring the smart viewcache cleaning system (optional)
5. Clearing the caches

Step 1: Upgrading the distribution files

The easiest way to upgrade the distribution files is to unpack eZ publish 3.4.8 to a directory and then copy the directories that contain site-specific files from the existing installation. Make sure that you copy the following directories:

- design/example
- design/example_admin
- var
- settings/siteaccess
- settings/override

Replace "example" and "example_admin" with actual names used by your siteaccesses.

Custom extensions

If you are using custom extensions then the subdirectories inside the "extension" directory will also have to be copied. However, make sure that you do not overwrite any extensions that come with eZ publish (for example the "PayPal" extension).

Step 2: Upgrading the database

To upgrade 3.4.0 database to 3.4.8, you should navigate into the eZ publish 3.4.8 directory and run the following database upgrade scripts one after another:

1. dbupdate-3.4.0-to-3.4.1.sql
2. dbupdate-3.4.1-to-3.4.2.sql
3. dbupdate-3.4.2-to-3.4.3.sql
4. dbupdate-3.4.3-to-3.4.4.sql
5. dbupdate-3.4.4-to-3.4.5.sql
6. dbupdate-3.4.5-to-3.4.6.sql
7. dbupdate-3.4.6-to-3.4.7.sql
8. dbupdate-3.4.7-to-3.4.8.sql

MySQL

The database upgrade scripts are located in the "update/database/mysql/3.4/" directory of your eZ publish installation. Each of these scripts can be launched using the following shell command:

```
mysql -u<username> -p<password> <database> < update/database/mysql/3.4/  
dbupdate-3.4.x-to-3.4.y.sql
```

PostgreSQL

The database upgrade scripts are located in the "update/database/postgresql/3.4/" directory of your eZ publish installation. Each of these scripts can be launched using the following shell command:

```
psql -d <database> -U <downer> < update/database/postgresql/3.4/  
dbupdate-3.4.x-to-3.4.y.sql
```

Step 3: Running the system upgrade scripts

There are no upgrade scripts for upgrading from 3.4.x to 3.4.y.

Step 4: Configuring the smart viewcache cleaning system (optional)

In eZ publish 3.4.2, the caching functionality has been improved (see the [3.4.2 changelog](#) for more information). If you are upgrading to version 3.4.2 or later, then it is recommended to configure the [smart viewcache cleaning system](#). This feature allows to determine which additional nodes the system should clear the caches for when the content of an object is changed (by default, the view cache will be cleared for all published nodes of this object, their parent nodes and nodes of reverse related objects). Please note that this step is only required if you are running a complex site or if you have cache problems after upgrading (for example, if your changes to content are not visible on some pages).

Step 5: Clearing the caches

Whenever an eZ publish solution is upgraded, all caches must be cleared in a proper way. This should be done from within a system shell:

1. Navigate into the eZ publish 3.4.8 directory.
2. Run the clear cache script:

```
bin/shell/clearcache.sh --clear-all
```

Please make sure that all caches are cleared. Sometimes the script is unable to clear caches because of restrictive file/directory permission settings. Make sure that all caches have been cleared by inspecting the contents of the various cache subdirectories within the "var" directory.

1.21.2 from 3.3-5 to 3.4.0

This section describes how to upgrade your existing eZ publish 3.3-5 installation to version 3.4.0. If you are upgrading from a version prior to eZ publish 3.3-5, you need to first upgrade to 3.3-5 (page 325) before you can upgrade to 3.4.0.

Please make sure that you have a working backup of the site before you do the actual upgrade. The upgrade procedure consists of the following steps:

1. Upgrading the distribution files to 3.4.0
2. Upgrading the database to 3.4.0
3. Running the 3.4.0 upgrade scripts
4. Updating the system configuration
5. Clearing the caches

Step 1: Upgrading the distribution files

The easiest way to upgrade the distribution files is to unpack eZ publish 3.4.0 to a directory and then copy the directories that contain site-specific files from the existing installation. Make sure that you copy the following directories:

- design/example
- design/example_admin
- var
- settings/siteaccess
- settings/override

Replace "example" and "example_admin" with actual names used by your siteaccesses.

Custom extensions

If you are using custom extensions then the subdirectories inside the "extension" directory will also have to be copied. However, make sure that you do not overwrite any extensions that come with eZ publish (for example the "PayPal" extension).

Step 2: Upgrading the database

The following text describes how a 3.3-5 database can be upgraded to 3.4.0.

MySQL

1. Navigate into the eZ publish 3.4.0 directory.
2. Run the database upgrade script:

```
mysql -u<username> -p<password> <database> < update/database/mysql/3.4/  
dbupdate-3.3-5-to-3.4.0.sql
```

PostgreSQL

1. Navigate into the eZ publish 3.4.0 directory.
2. Run the database upgrade script:

```
psql -d <database> -U <dbowner> < update/database/postgresql/3.4/  
dbupdate-3.3-5-to-3.4.0.sql
```

Step 3: Running the 3.4.0 upgrade scripts

The 3.4.0 version of eZ publish introduces a couple of new features. In order to make sure that your site is compatible with these features, you'll have to run a couple of upgrade scripts.

Remote ID

In eZ publish 3.4.0, a new attribute called "remote_id" has been added to content objects, classes and nodes. This attribute stores a global unique ID for an object/class/node. The remote ID is an MD5 hash of the time when the object/class/node was created plus a random value. These IDs are used in order to avoid collisions of identical objects/classes/nodes when importing and exporting data from one site to another. You need to run the "updateremoteid.php" script in order to make sure that all your items that support the remote ID gets one generated:

1. Navigate into the eZ publish 3.4.0 directory.
2. Run the script (replace "example" with the name of your siteaccess(es)):

```
php update/common/scripts/updateremoteid.php
```

Updating the session data

In eZ publish 3.4.0, the [session functionality](#) has been improved in order to relate a session to a user (see the [3.4.0 changelog](#) for more information). The session data in the "ezsession" table

now contain information about the session user. This allows to easily count the number of active registered users and anonymous users. When a user is being removed or disabled, the system will remove the corresponding session(s) in order to automatically log out invalid users. In order to update the entries in the "ezsession" table, you need to run the "updatesession.php" script:

1. Navigate into the eZ publish 3.4.0 directory.
2. Run the script:

```
php update/common/scripts/updatesession.php
```

Top level nodes/objects changes

In eZ publish 3.4.0, a new top level node/object called "Setup" has been added. You need to run the "updatetoplevel.php" script for your admin siteaccess in order to make sure that all your top level nodes are properly configured (replace "example_admin" with the actual name of your admin siteaccess):

1. Navigate into the eZ publish 3.4.0 directory.
2. Run the script:

```
php update/common/scripts/updatetoplevel.php
```

The script will check the top level nodes and match them with the values defined under the "[NodeSettings]" section of the "content.ini" configuration file or its override. If "content.ini" or its override contains incorrect IDs of the top level nodes, these will be automatically updated. If some of the top level nodes are missing, these will be created.

Changes in the shop order system

In eZ publish 3.4.0, the webshop functionality has been improved (see the [3.4.0 changelog](#) for more information). The list of orders that is stored in the "ezorder" database table now includes the customer's email for each order. If you are using the eZ publish webshop functionality on your site and your order list is not empty, then you need to run the "addorderemail.php" script when upgrading to 3.4.0:

1. Navigate into the eZ publish 3.4.0 directory.
2. Run the script:

```
php update/common/scripts/addorderemail.php
```

The script will modify the "ezorder" table in the database.

Step 4: Updating the system configuration

You are not required to do any configuration changes when upgrading from 3.3-5 to 3.4.0. However, please note that it is not recommended to use "admin" as a name of your administration siteaccess. This name is reserved by the system and your changes to the "admin" siteaccess will be lost during the upgrade process because of overwriting the files in the "settings/siteaccess/admin" directory. Trying to use the "admin" siteaccess may cause problems when upgrading. A typical symptom is that everything is working fine after upgrading, except for some parts of your administration interface, e.g. media and user tabs do not work. To solve this, create a new siteaccess called for example "my_admin" and copy the altered files from your backup to "settings/siteaccess/my_admin".

Step 5: Clearing the caches

Whenever an eZ publish solution is upgraded, all caches must be cleared in a proper way. This should be done from within a system shell:

1. Navigate into the eZ publish 3.4.0 directory.
2. Run the clear cache script:

```
bin/shell/clearcache.sh --clear-all
```

Please make sure that all caches are cleared. Sometimes the script is unable to clear caches because of restrictive file/directory permission settings. Make sure that all caches have been cleared by inspecting the contents of the various cache subdirectories within the "var" directory.

1.22 Upgrading to 3.3

This chapter contains information related to the upgrade of an eZ Publish solution to [version 3.3](#). Please follow the upgrade procedures that are described in the following sections:

- [Upgrading from 3.3-x to 3.3-y \(page 325\)](#)
- [Upgrading from 3.2-4 to 3.3-1 \(page 328\)](#)

1.22.1 from 3.3-x to 3.3-y

This section describes how to upgrade your existing eZ Publish 3.3-x installation to version 3.3-y, for example from 3.3-1 to 3.3-7. If you are upgrading from a version prior to eZ Publish 3.3-1, you should first upgrade to 3.3-1 (page [328](#)) before you can upgrade to 3.3-7.

Please make sure that you have a working backup of the site before you do the actual upgrade. The upgrade procedure consists of the following steps:

1. Upgrading the distribution files to 3.3-7
2. Upgrading the database to 3.3-7
3. Running the system upgrade scripts
4. Updating the system configuration
5. Clearing the caches

Step 1: Upgrading the distribution files

The easiest way to upgrade the distribution files is to unpack eZ Publish 3.3-7 to a directory and then copy the directories that contain site-specific files from the existing installation. Make sure that you copy the following directories:

- design/example
- design/example_admin
- var
- settings/siteaccess
- settings/override

Replace "example" and "example_admin" with actual names used by your siteaccesses.

Custom extensions

If you are using custom extensions then the subdirectories inside the "extension" directory will also have to be copied. However, make sure that you do not overwrite any extensions that come with eZ Publish (for example the "PayPal" extension).

Step 2: Upgrading the database

To upgrade 3.3-1 database to 3.3-7, you should navigate into the eZ Publish 3.3-7 directory and run the following database upgrade scripts one after another:

1. dbupdate-3.3-1-to-3.3-2.sql
2. dbupdate-3.3-2-to-3.3-3.sql
3. dbupdate-3.3-3-to-3.3-4.sql
4. dbupdate-3.3-4-to-3.3-5.sql
5. dbupdate-3.3-5-to-3.3-6.sql
6. dbupdate-3.3-6-to-3.3-7.sql

MySQL

The database upgrade scripts are located in the "update/database/mysql/3.3/" directory of your eZ Publish installation. Each of these scripts can be launched using the following shell command:

```
mysql -u<username> -p<password> <database> < update/database/mysql/3.3/  
dbupdate-3.3-x-to-3.3-y.sql
```

PostgreSQL

The database upgrade scripts are located in the "update/database/postgresql/3.3/" directory of your eZ Publish installation. Each of these scripts can be launched using the following shell command:

```
psql -d <database> -U <downer> < update/database/postgresql/3.3/  
dbupdate-3.3-x-to-3.3-y.sql
```

Step 3: Running the system upgrade scripts

There are no upgrade scripts for upgrading from 3.3-x to 3.3-y.

Step 4: Updating the system configuration

You are not required to do any configuration changes when upgrading from 3.3-x to 3.3-y.

Step 5: Clearing the caches

Whenever an eZ Publish solution is upgraded, all caches must be cleared in a proper way. This should be done from within a system shell:

1. Navigate into the eZ Publish 3.3-7 directory.
2. Run the clear cache script:

```
bin/shell/clearcache.sh --clear-all
```

Please make sure that all caches are cleared. Sometimes the script is unable to clear caches because of restrictive file/directory permission settings. Make sure that all caches have been cleared by inspecting the contents of the various cache subdirectories within the "var" directory.

1.22.2 from 3.2-4 to 3.3-1

This section describes how to upgrade your existing eZ Publish 3.2-4 installation to version 3.3-1. If you are upgrading from a version prior to eZ Publish 3.2-4, you need to first upgrade to 3.2-4 (page 331) before you can upgrade to 3.3-1.

Please make sure that you have a working backup of the site before you do the actual upgrade. The upgrade procedure consists of the following steps:

1. Upgrading the distribution files to 3.3-1
2. Upgrading the database to 3.3-1
3. Running the system upgrade scripts
4. Updating the system configuration
5. Clearing the caches

Step 1: Upgrading the distribution files

The easiest way to upgrade the distribution files is to unpack eZ Publish 3.3-1 to a directory and then copy the directories that contain site-specific files from the existing installation. Make sure that you copy the following directories:

- design/example
- design/example_admin
- var
- settings/siteaccess
- settings/override

Replace "example" and "example_admin" with actual names used by your siteaccesses.

Custom extensions

If you are using custom extensions then the subdirectories inside the "extension" directory will also have to be copied. However, make sure that you do not overwrite any extensions that come with eZ Publish (for example the "PayPal" extension).

Step 2: Upgrading the database

The following text describes how a 3.2-4 database can be upgraded to 3.3-1.

MySQL

1. Navigate into the eZ Publish 3.3-1 directory.
2. Run the database upgrade script:

```
mysql -u<username> -p<password> <database> < update/database/mysql/3.3/  
dbupdate-3.2-4-to-3.3-1.sql
```

PostgreSQL

1. Navigate into the eZ Publish 3.3-1 directory.
2. Run the database upgrade script:

```
psql -d <database> -U <dbowner> < update/database/postgresql/3.3/  
dbupdate-3.2-4-to-3.3-1.sql
```

Step 3: Running the system upgrade scripts

There are no upgrade scripts for upgrading from 3.2-4 to 3.3-1.

Step 4: Updating the system configuration

You are not required to do any configuration changes when upgrading from 3.2-4 to 3.3-1.

Step 5: Clearing the caches

Whenever an eZ Publish solution is upgraded, all caches must be cleared in a proper way. This should be done from within a system shell:

1. Navigate into the eZ Publish 3.3-1 directory.
2. Run the clear cache script:

```
bin/shell/clearcache.sh --clear-all
```

Please make sure that all caches are cleared. Sometimes the script is unable to clear caches because of restrictive file/directory permission settings. Make sure that all caches have been cleared by inspecting the contents of the various cache subdirectories within the "var" directory.

1.23 Upgrading to 3.2

This chapter contains information related to the upgrade of an eZ Publish solution to [version 3.2](#). Please follow the upgrade procedures that are described in the following sections:

- [Upgrading from 3.2-x to 3.2-y \(page 331\)](#)
- [Upgrading from 3.1-1 to 3.2-1 \(page 334\)](#)

1.23.1 from 3.2-x to 3.2-y

This section describes how to upgrade your existing eZ Publish 3.2-x installation to version 3.2-y, for example from 3.2-1 to 3.2-6. If you are upgrading from a version prior to eZ Publish 3.2-1, you should first upgrade to 3.2-1 (page [334](#)) before you can upgrade to 3.2-6.

Please make sure that you have a working backup of the site before you do the actual upgrade. The upgrade procedure consists of the following steps:

1. Upgrading the distribution files to 3.2-6
2. Upgrading the database to 3.2-6
3. Creating missing translations (optional)
4. Updating the system configuration
5. Clearing the caches

Step 1: Upgrading the distribution files

The easiest way to upgrade the distribution files is to unpack eZ Publish 3.2-6 to a directory and then copy the directories that contain site-specific files from the existing installation. Make sure that you copy the following directories:

- design/example
- design/example_admin
- var
- settings/siteaccess
- settings/override

Replace "example" and "example_admin" with actual names used by your siteaccesses.

Custom extensions

If you are using custom extensions then the subdirectories inside the "extension" directory will also have to be copied. However, make sure that you do not overwrite any extensions that come with eZ Publish (for example the "PayPal" extension).

Step 2: Upgrading the database

To upgrade 3.2-1 database to 3.2-6, you should navigate into the eZ Publish 3.2-6 directory and run the following database upgrade scripts one after another:

1. dbupdate-3.2-1-to-3.2-2.sql
2. dbupdate-3.2-2-to-3.2-3.sql
3. dbupdate-3.2-3-to-3.2-4.sql
4. dbupdate-3.2-4-to-3.2-5.sql
5. dbupdate-3.2-5-to-3.2-6.sql

MySQL

The database upgrade scripts are located in the "update/database/mysql/3.2/" directory of your eZ Publish installation. Each of these scripts can be launched using the following shell command:

```
mysql -u<username> -p<password> <database> < update/database/mysql/3.2/  
dbupdate-3.2-x-to-3.2-y.sql
```

PostgreSQL

The database upgrade scripts are located in the "update/database/postgresql/3.2/" directory of your eZ Publish installation. Each of these scripts can be launched using the following shell command:

```
psql -d <database> -U <downer> < update/database/postgresql/3.2/  
dbupdate-3.2-x-to-3.2-y.sql
```

Step 3: Creating missing translations (optional)

In eZ Publish 3.2-4, an important [bug](#) related to adding new object attributes has been fixed (see the [3.2-4 changelog](#) for more information). Since that bug might lead to missing translated attributes on multi-lingual sites, it is recommended to run the "updatetranslations.php" script when upgrading to 3.2-4 or later version. The script will check your content objects and re-initialize attributes if their data is missing as a result of the bug.

Step 4: Updating the system configuration

You are not required to do any configuration changes when upgrading from 3.2-x to 3.2-y.

Step 5: Clearing the caches

Whenever an eZ Publish solution is upgraded, all caches must be cleared in a proper way. This should be done from within a system shell:

1. Navigate into the eZ Publish 3.2-6 directory.
2. Run the clear cache script:

```
bin/shell/clearcache.sh --clear-all
```

Please make sure that all caches are cleared. Sometimes the script is unable to clear caches because of restrictive file/directory permission settings. Make sure that all caches have been cleared by inspecting the contents of the various cache subdirectories within the "var" directory.

1.23.2 from 3.1-1 to 3.2-1

This section describes how to upgrade your existing eZ Publish 3.1-1 installation to version 3.2-1. If you are upgrading from a version prior to eZ Publish 3.1-1, you need to first upgrade to 3.1-1 (page 338) before you can upgrade to 3.2-1.

Please make sure that you have a working backup of the site before you do the actual upgrade. The upgrade procedure consists of the following steps:

1. Upgrading the distribution files to 3.2-1
2. Upgrading the database to 3.2-1
3. Running the system upgrade scripts
4. Updating the system configuration
5. Clearing the caches

Step 1: Upgrading the distribution files

The easiest way to upgrade the distribution files is to unpack eZ Publish 3.2-1 to a directory and then copy the directories that contain site-specific files from the existing installation. Make sure that you copy the following directories:

- design/example
- design/example_admin
- var
- settings/siteaccess
- settings/override

Replace "example" and "example_admin" with actual names used by your siteaccesses.

Custom extensions

If you are using custom extensions then the subdirectories inside the "extension" directory will also have to be copied. However, make sure that you do not overwrite any extensions that come with eZ Publish (for example the "PayPal" extension).

Step 2: Upgrading the database

The following text describes how a 3.1-1 database can be upgraded to 3.2-1.

MySQL

1. Navigate into the eZ Publish 3.2-1 directory.
2. Run the database upgrade script:

```
mysql -u<username> -p<password> <database> < update/database/mysql/3.2/  
dbupdate-3.1-1-to-3.2-1.sql
```

PostgreSQL

1. Navigate into the eZ Publish 3.2-1 directory.
2. Run the database upgrade script:

```
psql -d <database> -U <dbowner> < update/database/postgresql/3.2/  
dbupdate-3.1-1-to-3.2-1.sql
```

Step 3: Running the system upgrade scripts

eZ Publish 3.2-1 supports sorting and filtering on attribute level. It is possible to sort and/or filter a list of content objects based on the values of a specific attribute (see the [3.2-1 changelog](#) for more information). The "list" and "tree" fetch functions support this functionality. In order to be compatible with this new feature, you need to run the "updateattributefilter.php" script when upgrading:

```
php update/common/scripts/updateattributefilter.php
```

In addition, if you haven't run the "generateoverridesettings.php" script before (when upgrading to 3.1.0 rel1 or later), then it needs to be run for each of your siteaccesses when upgrading to 3.2-1 (replace "example" with the actual name of your siteaccess):

```
php update/common/scripts/generateoverridesettings.php -s example
```

The script will output a suggestion for the siteaccess "override.ini.append.php" file that needs to be created for the specified siteaccess. You should create a new file called "override.ini.append.php" in the "settings/siteaccess/example" directory (replace "example" by the actual name of your siteaccess) and copy the output provided by the script to this file.

Step 4: Updating the system configuration

You are not required to do any configuration changes when upgrading from 3.1-1 to 3.2-1.

Step 5: Clearing the caches

Whenever an eZ Publish solution is upgraded, all caches must be cleared in a proper way. This should be done from within a system shell:

1. Navigate into the eZ Publish 3.2-1 directory.
2. Run the clear cache script:

```
bin/shell/clearcache.sh --clear-all
```

Please make sure that all caches are cleared. Sometimes the script is unable to clear caches because of restrictive file/directory permission settings. Make sure that all caches have been cleared by inspecting the contents of the various cache subdirectories within the "var" directory.

1.24 Upgrading to 3.1

This chapter contains information related to the upgrade of an eZ Publish solution to [version 3.1](#). Please follow the upgrade procedures that are described in the following sections:

- [Upgrading from 3.1-x to 3.1-y \(page 338\)](#)
- [Upgrading from 3.0-2 to 3.1.0 rel1 \(page 341\)](#)

1.24.1 from 3.1-x to 3.1-y

This section describes how to upgrade your existing eZ Publish 3.1-x installation to version 3.1-y, for example from 3.1.0 rel1 to 3.1-2. If you are upgrading from a version prior to eZ Publish 3.1.0 rel1, you should first upgrade to 3.1.0 rel1 (page [341](#)) before you can upgrade to 3.1-2.

Please make sure that you have a working backup of the site before you do the actual upgrade. The upgrade procedure consists of the following steps:

1. Upgrading the distribution files to 3.1-2
2. Upgrading the database to 3.1-2
3. Running the system upgrade scripts
4. Updating the system configuration
5. Clearing the caches

Step 1: Upgrading the distribution files

The easiest way to upgrade the distribution files is to unpack eZ Publish 3.1-2 to a directory and then copy the directories that contain site-specific files from the existing installation. Make sure that you copy the following directories:

- design/example
- design/example_admin
- var
- settings/siteaccess
- settings/override

Replace "example" and "example_admin" with actual names used by your siteaccesses.

Custom extensions

If you are using custom extensions then the subdirectories inside the "extension" directory will also have to be copied. However, make sure that you do not overwrite any extensions that come with eZ Publish (for example the "PayPal" extension).

Step 2: Upgrading the database

To upgrade 3.1.0 rel1 database to 3.1-2, you should navigate into the eZ Publish 3.1-2 directory and run the following database upgrade scripts one after another:

1. dbupdate-3.1.0-1-to-3.1.0-2.sql
2. dbupdate-3.1.0-2-to-3.1-1.sql

MySQL

The database upgrade scripts are located in the "update/database/mysql/3.1/" directory of your eZ Publish installation. Each of these scripts can be launched using the following shell command:

```
mysql -u<username> -p<password> <database> < update/database/mysql/3.1/  
dbupdate-version1-to-version2.sql
```

PostgreSQL

The database upgrade scripts are located in the "update/database/postgresql/3.1/" directory of your eZ Publish installation. Each of these scripts can be launched using the following shell command:

```
psql -d <database> -U <dbowner> < update/database/postgresql/3.1/  
dbupdate-version1-to-version2.sql
```

Step 3: Running the system upgrade scripts

There are no upgrade scripts for upgrading from 3.1-x to 3.1-y. However, if you haven't run the "generateoverridesettings.php" script before (when upgrading to 3.1.0 rel1), then it needs to be run for each of your siteaccesses (replace "example" with the actual name of your siteaccess):

```
php update/common/scripts/generateoverridesettings.php -s example
```

The script will output a suggestion for the siteaccess "override.ini.append.php" file that needs to be created for the specified siteaccess. You should create a new file called "override.ini.append.php" in the "settings/siteaccess/example" directory (replace "example" by the actual name of your siteaccess) and copy the output provided by the script to this file.

Step 4: Updating the system configuration

You are not required to do any configuration changes when upgrading from 3.1-x to 3.1-y.

Step 5: Clearing the caches

Whenever an eZ Publish solution is upgraded, all caches must be cleared in a proper way. This should be done from within a system shell:

1. Navigate into the eZ Publish 3.1-2 directory.
2. Run the clear cache script:

```
bin/shell/clearcache.sh --clear-all
```

Please make sure that all caches are cleared. Sometimes the script is unable to clear caches because of restrictive file/directory permission settings. Make sure that all caches have been cleared by inspecting the contents of the various cache subdirectories within the "var" directory.

1.24.2 from 3.0-2 to 3.1.0 rel1

This section describes how to upgrade your existing eZ Publish 3.0-2 installation to version 3.1.0 rel1. If you are upgrading from a version prior to eZ Publish 3.0-2, you need to first upgrade to 3.0-2 (page 345) before you can upgrade to 3.1.0 rel1.

Please make sure that you have a working backup of the site before you do the actual upgrade. The upgrade procedure consists of the following steps:

1. Upgrading the distribution files to 3.1.0 rel1
2. Upgrading the database to 3.1.0 rel1
3. Generating override settings
4. Updating the system configuration
5. Clearing the caches

Step 1: Upgrading the distribution files

The easiest way to upgrade the distribution files is to unpack eZ Publish 3.1.0 rel1 to a directory and then copy the directories that contain site-specific files from the existing installation. Make sure that you copy the following directories:

- design/example
- design/example_admin
- var
- settings/siteaccess
- settings/override

Replace "example" and "example_admin" with actual names used by your siteaccesses.

Custom extensions

If you are using custom extensions then the subdirectories inside the "extension" directory will also have to be copied. However, make sure that you do not overwrite any extensions that come with eZ Publish (for example the "PayPal" extension).

Step 2: Upgrading the database

The following text describes how a 3.0-2 database can be upgraded to 3.1.0 rel1.

MySQL

1. Navigate into the eZ Publish 3.1.0 rel1 directory.
2. Run the database upgrade script:

```
mysql -u<username> -p<password> <database> < update/database/mysql/3.1/  
dbupdate-3.0-2-to-3.1.0-1.sql
```

PostgreSQL

1. Navigate into the eZ Publish 3.1.0 rel1 directory.
2. Run the database upgrade script:

```
psql -d <database> -U <dbowner> < update/database/postgresql/3.1/  
dbupdate-3.0-2-to-3.1.0-1.sql
```

Step 3: Generating override settings

In eZ Publish 3.1.0 rel1, the new template override subsystem that allows to use custom templates instead of the default/standard ones has been implemented (see the [3.1.0 rel1 changelog](#) for more information). The template overrides must be defined in the "override.ini.append.php" file of a siteaccess. In order to be compatible with this possibility, you will need to run the "generateoverridesettings.php" script for each of your siteaccesses (replace "example" with the actual name of your siteaccess):

```
php update/common/scripts/generateoverridesettings.php -s example
```

The script will output a suggestion for the siteaccess "override.ini.append.php" file that needs to be created for the specified siteaccess. You should create a new file called "override.ini.append.php" in the "settings/siteaccess/example" directory (replace "example" by the actual name of your siteaccess) and copy the output provided by the script to this file.

Step 4: Updating the system configuration

You are not required to do any configuration changes when upgrading from 3.0-2 to 3.1.0 rel1.

Step 5: Clearing the caches

Whenever an eZ Publish solution is upgraded, all caches must be cleared in a proper way. This should be done from within a system shell:

1. Navigate into the eZ Publish 3.1.0 rel1 directory.

2. Run the clear cache script:

```
bin/shell/clearcache.sh --clear-all
```

Please make sure that all caches are cleared. Sometimes the script is unable to clear caches because of restrictive file/directory permission settings. Make sure that all caches have been cleared by inspecting the contents of the various cache subdirectories within the "var" directory.

1.25 Upgrading to 3.0

This chapter contains information related to the upgrade of an eZ Publish solution to [version 3.0](#). Please follow the upgrade procedures that are described in the following sections:

- [Upgrading from 3.0-1 to 3.0-2 \(page 345\)](#)
- [Upgrading from 2.9.7 to 3.0-1 \(page 347\)](#)

1.25.1 from 3.0-1 to 3.0-2

This section describes how to upgrade your existing eZ Publish 3.0-1 installation to version 3.0-2. If you are upgrading from 2.9.x, then you should first upgrade to 3.0-1 (page 347) before you can upgrade to 3.0-2.

Please make sure that you have a working backup of the site before you do the actual upgrade. The upgrade procedure consists of the following steps:

1. Upgrading the distribution files to 3.0-2
2. Upgrading the database to 3.0-2
3. Running the system upgrade scripts
4. Updating the system configuration
5. Clearing the caches

Step 1: Upgrading the distribution files

The easiest way to upgrade the distribution files is to unpack eZ Publish 3.0-2 to a directory and then copy the directories that contain site-specific files from the existing installation. Make sure that you copy the following directories:

- design/example
- design/example_admin
- var
- settings/siteaccess
- settings/override

Replace "example" and "example_admin" with actual names used by your siteaccesses.

Custom extensions

If you are using custom extensions then the subdirectories inside the "extension" directory will also have to be copied. However, make sure that you do not overwrite any extensions that come with eZ Publish (for example the "PayPal" extension).

Step 2: Upgrading the database

There is no need to upgrade 3.0-1 database to 3.0-2 (the appropriate database upgrade script is empty).

Step 3: Running the system upgrade scripts

There are no upgrade scripts for upgrading from 3.0-1 to 3.0-2.

Step 4: Updating the system configuration

You are not required to do any configuration changes when upgrading from 3.0-1 to 3.0-2.

Step 5: Clearing the caches

Whenever an eZ Publish solution is upgraded, all caches must be cleared in a proper way. This should be done from within a system shell:

1. Navigate into the eZ Publish 3.0-2 directory.
2. Run the clear cache script:

```
bin/shell/clearcache.sh --clear-all
```

Please make sure that all caches are cleared. Sometimes the script is unable to clear caches because of restrictive file/directory permission settings. Make sure that all caches have been cleared by inspecting the contents of the various cache subdirectories within the "var" directory.

1.25.2 from 2.9-7 to 3.0-1

This section describes how to upgrade your existing eZ Publish 2.9-7 installation to version 3.0-1. If you are upgrading from a version prior to eZ Publish 2.9-7, you need to first upgrade to 2.9-7 before you can upgrade to 3.0-1.

Please make sure that you have a working backup of the site before you do the actual upgrade. The upgrade procedure consists of the following steps:

1. Upgrading the distribution files to 3.0-1
2. Upgrading the database to 3.0-1
3. Running the 3.0-1 upgrade scripts
4. Updating the system configuration
5. Clearing the caches

Step 1: Upgrading the distribution files

The easiest way to upgrade the distribution files is to unpack eZ Publish 3.0-1 to a directory and then copy the directories that contain site-specific files from the existing installation. Make sure that you copy the following directories:

- design/example
- design/example_admin
- var
- settings/siteaccess
- settings/override

Replace "example" and "example_admin" with actual names used by your siteaccesses.

Custom extensions

If you are using custom extensions then the subdirectories inside the "extension" directory will also have to be copied. However, make sure that you do not overwrite any extensions that come with eZ Publish (for example the "PayPal" extension).

Step 2: Upgrading the database

The following text describes how a 2.9-7 database can be upgraded to 3.0-1.

MySQL

1. Navigate into the eZ Publish 3.0-1 directory.
2. Run the database upgrade script:

```
mysql -u<username> -p<password> <database> < update/database/mysql/3.0/  
dbupdate-2.9-7-to-3.0-1.sql
```

PostgreSQL

1. Navigate into the eZ Publish 3.0-1 directory.
2. Run the database upgrade script:

```
psql -d <database> -U <dbowner> < update/database/postgresql/3.0/  
dbupdate-2.9-7-to-3.0-1.sql
```

Step 3: Running the 3.0-1 upgrade scripts

The 3.0-1 version of eZ Publish introduces a couple of new features. In order to make sure that your site is compatible with these feature, you'll have to run a couple of upgrade scripts.

Updating nice URLs

In eZ Publish 3.0, the virtual URLs (nice URLs, URL aliases) are automatically generated and maintained for all nodes. The generated URL for a node is based on the node's location in the tree and the actual name of the object that the node encapsulates. In order to be compatible with this feature, you will need to run the "updateniceurls.php" script:

1. Navigate into the eZ Publish 3.0-1 directory.
2. Run the script using the following shell command:

```
php update/common/scripts/updateniceurls.php
```

The script will generate virtual URLs for all your content nodes.

Updating XML fields

In eZ Publish 3.0-1, the "ezxmltext" datatype has been changed in order to store an XML code using UTF-8 encoding (see the [3.0-1 changelog](#) for more information). In order to be compatible with this feature, you will need to run the "updatexmltext.php" script:

1. Navigate into the eZ Publish 3.0-1 directory.
2. Run the script using the following shell command:

```
php update/common/scripts/updatesxmltext.php
```

The script will automatically convert your XML fields to new format. It is recommended to create a database backup before using this script.

Updating the search index data

In eZ Publish 3.0-1, the built-in search engine has been improved (see the [3.0-1 changelog](#) for more information) and thus you will need to update the search index of your site when upgrading. To do this, clear the search index tables in your database (these are called "ezsearch_word" and "ezsearch_object_word_link") and run the "updatesearchindex.php" script:

1. Navigate into the eZ Publish 3.0-1 directory.
2. Run the script using the following shell command:

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

Step 4: Updating the system configuration

You are not required to do any configuration changes when upgrading from 2.9-7 to 3.0-1.

Step 5: Clearing the caches

Whenever an eZ Publish solution is upgraded, all caches must be cleared in a proper way. This should be done from within a system shell:

1. Navigate into the eZ Publish 3.0-1 directory.
2. Run the clear cache script:

```
bin/shell/clearcache.sh --clear-all
```

Please make sure that all caches are cleared. Sometimes the script is unable to clear caches because of restrictive file/directory permission settings. Make sure that all caches have been cleared by inspecting the contents of the various cache subdirectories within the "var" directory.

1.26 Upgrading from 3.a.b to 3.x.y

This section describes how to upgrade your existing eZ Publish 3.a.b installation to version 3.x.y. In order to benefit from the latest security and bug fixes, choose the latest stable release in the desired branch.

If version 3.a.b is a few branches older than version 3.x.y, split the upgrade process into several stages. This means that you should upgrade directly to the next stable release branch, and then to the next one, etc. until you reach the desired version. While the last direct upgrade in the chain should be done to version 3.x.y, there are two different rules about which intermediary versions to choose for preceding upgrades. You can either follow the official upgrade path for database schema changes or select the latest stable release in each branch. Refer to the "How to proceed (page 11)" section for more information and examples.

The following text explains how to do a direct upgrade to the next stable release branch. In addition, the instructions below can be used to upgrade directly from eZ Publish 3.6 to version 3.8.

The procedure for directly upgrading from version 3.a.b to 3.x.y consists of the following steps:

1. Upgrading the distribution files to version 3.x.y.
2. Upgrading the database schema to version 3.x.y.
3. Running the system upgrade scripts for versions from 3.a.b to 3.x.y.
4. Updating the system configuration for compatibility with all the changes made from version 3.a.b to 3.x.y.
5. Clearing the caches.

Make sure that you have a working backup of the site before you do the actual upgrade.

Step 1: Upgrading the distribution files

The easiest way to upgrade the distribution files is to unpack eZ Publish 3.x.y to a separate directory and then copy the directories that contain site-specific files from the existing installation. Make sure that you copy the following directories:

- design/example
- design/example_admin
- var
- settings/siteaccess
- settings/override

Replace "example" and "example_admin" with the actual names of your siteaccesses. (If you have more than two siteaccesses, copy the design subdirectories for the others as well.)

Custom extensions

If you are using custom extensions, the subdirectories inside the "extension" directory will also have to be copied. However, make sure that you do not overwrite any extensions that come with eZ Publish (for example the "PayPal" extension).

Version specific notes

There may be some additional notes about upgrading the distribution files for each particular version of eZ Publish. For example, you may need to upgrade some extensions for compatibility with eZ Publish 3.x.y or download additional files. Read the first step of the upgrade manuals for the target branch and make sure you do not miss anything important.

For example, when upgrading from 3.6 to 3.8.10, you need to quickly skim through the first step of the "Upgrading from 3.6.x or 3.7.x to 3.8.0" and "Upgrading from 3.8.x to 3.8.y (page 280)" manuals. You will see that the latter one contains an important note about downloading an additional file in order to make your user registration mechanism work properly in 3.8.10.

Step 2: Upgrading the database

New eZ Publish releases usually include database changes. This means that the old database needs to be converted. To upgrade an eZ Publish database from version 3.a.b to 3.x.y, navigate into the eZ Publish 3.x.y directory and run the database upgrade scripts one after another following the official upgrade path. The upgrade scripts are located under the "update/database/mysql" directory (or "update/database/postgresql" if you are using PostgreSQL).

The following table shows examples for upgrading an eZ Publish database to different versions. Note that some scripts can be skipped since they do not do anything except update the version number (these are encapsulated by square brackets in the table).

Version	Database upgrade scripts to run
from 2.9-7 to 3.0-2	.../3.0/dbupdate-2.9.7-to-3.0-1.sql
from 3.0-2 to 3.1-1	.../3.1/dbupdate-3.0-2-to-3.1.0-1.sql .../3.1/dbupdate-3.1.0-1-to-3.1.0-2.sql .../3.1/dbupdate-3.1.0-2-to-3.1-1.sql
from 3.1-1 to 3.2-4	.../3.2/dbupdate-3.1-1-to-3.2-1.sql .../3.2/dbupdate-3.2-1-to-3.2-2.sql [.../3.2/dbupdate-3.2-2-to-3.2-3.sql] .../3.2/dbupdate-3.2-3-to-3.2-4.sql
from 3.2-4 to 3.3-5	.../3.3/dbupdate-3.2-4-to-3.3-1.sql .../3.3/dbupdate-3.3-1-to-3.3-2.sql .../3.3/dbupdate-3.3-2-to-3.3-3.sql [.../3.3/dbupdate-3.3-3-to-3.3-4.sql] .../3.3/dbupdate-3.3-4-to-3.3-5.sql
from 3.3-5 to 3.4.4	.../3.4/dbupdate-3.3.5-to-3.4.0.sql .../3.4/dbupdate-3.4.0-to-3.4.1.sql

	.../3.4/dbupdate-3.4.1-to-3.4.2.sql .../3.4/dbupdate-3.4.2-to-3.4.3.sql .../3.4/dbupdate-3.4.3-to-3.4.4.sql
from 3.4.4 to 3.5.2	.../3.5/dbupdate-3.4.4-to-3.5.0.sql .../3.5/dbupdate-3.5.0-to-3.5.1.sql .../3.5/dbupdate-3.5.1-to-3.5.2.sql
from 3.5.2 to 3.6.12	.../3.6/dbupdate-3.5.2-to-3.6.0.sql .../3.6/dbupdate-3.6.0-to-3.6.1.sql [.../3.6/dbupdate-3.6.1-to-3.6.2.sql] [.../3.6/dbupdate-3.6.2-to-3.6.3.sql] [.../3.6/dbupdate-3.6.3-to-3.6.4.sql] [.../3.6/dbupdate-3.6.4-to-3.6.5.sql] [.../3.6/dbupdate-3.6.5-to-3.6.6.sql] [.../3.6/dbupdate-3.6.6-to-3.6.7.sql] [.../3.6/dbupdate-3.6.7-to-3.6.8.sql] .../3.6/dbupdate-3.6.8-to-3.6.9.sql [.../3.6/dbupdate-3.6.9-to-3.6.10.sql] [.../3.6/dbupdate-3.6.10-to-3.6.11.sql] .../3.6/dbupdate-3.6.11-to-3.6.12.sql
from 3.6 to 3.8.10	.../3.8/dbupdate-3.6.0-to-3.8.0.sql .../3.8/dbupdate-3.8.0-to-3.8.1.sql [.../3.8/dbupdate-3.8.1-to-3.8.2.sql] [.../3.8/dbupdate-3.8.2-to-3.8.3.sql] [.../3.8/dbupdate-3.8.3-to-3.8.4.sql] .../3.8/dbupdate-3.8.4-to-3.8.5.sql [.../3.8/dbupdate-3.8.5-to-3.8.6.sql] [.../3.8/dbupdate-3.8.6-to-3.8.7.sql] [.../3.8/dbupdate-3.8.7-to-3.8.8.sql] [.../3.8/dbupdate-3.8.8-to-3.8.9.sql] .../3.8/dbupdate-3.8.9-to-3.8.10.sql
from 3.8 to 3.9.4	.../3.8/dbupdate-3.8.0-to-3.9.0.sql .../3.8/dbupdate-3.9.0-to-3.9.1.sql [.../3.8/dbupdate-3.9.1-to-3.9.2.sql] .../3.8/dbupdate-3.9.2-to-3.9.3.sql .../3.8/dbupdate-3.9.3-to-3.9.4.sql

MySQL

A database upgrade script can be launched using the following shell command:

```
mysql -uUSERNAME -p[PASSWORD] DATABASE < update/database/mysql/BRANCH/
dbupdate-VERSION1-to-VERSION2.sql
```

for example:

```
mysql -umyuser -mypass mydatabase < update/database/mysql/3.5/  
dbupdate-3.4.4-to-3.5.0.sql
```

Note that the CREATE TABLE statements in the database upgrade scripts for eZ Publish 3 do not specify which storage engine to use (no ENGINE or TYPE option), and thus the default storage engine will be used. Normally, it is MyISAM (starting from MySQL v.3.23). If you are using InnoDB, make sure the default storage engine is set to InnoDB before you run the database upgrade scripts (refer to the [MySQL documentation](#) for information about how to set the default engine). If you were not able to change the MySQL configuration on your server, and the upgrade left you with a mix of table types, you can use the "bin/php/ezconvertmysqltabletype.php" script for database conversion. The script is available in eZ Publish 3.6.0 and later versions.

It is also possible to convert the newly created tables to InnoDB using ALTER TABLE statements as shown in the following example:

```
ALTER TABLE table_name1 TYPE = innodb;  
ALTER TABLE table_name2 TYPE = innodb;  
...
```

Replace "table_name1", "table_name2" with the actual names of the tables that need to be converted.

PostgreSQL

A database upgrade script can be launched using the following shell command:

```
psql -d DATABASE -U DBOWNER < update/database/postgresql/BRANCH/  
dbupdate-VERSION1-to-VERSION2.sql
```

for example:

```
psql -d mydatabase -U myuser < update/database/postgresql/3.5/  
dbupdate-3.4.4-to-3.5.0.sql
```

Step 3: Running the system upgrade scripts

A new release often introduces some new functionality that requires your existing data (like content classes, objects, nodes, search index etc.) to be updated. When upgrading, you may need to run some upgrade scripts in order to make your site compatible with the new features added between 3.a.b and 3.x.y. You will find most of the system upgrade scripts in the "update/common/scripts" subdirectory of the eZ Publish 3.x.y directory. To find out which scripts need to be run, view the list of the system upgrade scripts (page [357](#)).

The following table contains examples showing which system upgrade scripts to run when upgrading to the different versions.

Version	Scripts to run	Where to look for instructions
from 2.9-7 to 3.0-2	updateniceurls.php updatesearchindex.php updatexmltext.php	List of scripts (page 357) Upgrading from 2.9.7 to 3.0-1 (page 347)
from 3.3-5 to 3.4.4	updateremoteid.php updatesession.php updatetoplevel.php addorderemail.php	List of scripts (page 357) Upgrading from 3.3-5 to 3.4.0 (page 320)
from 3.4.4 to 3.5.11	updateiscontainer.php updatesearchindex.php updateniceurls.php updatetoplevel.php (added in 3.5.1) updateeztimetype.php (added in 3.5.3) updatecrc32.php (added in 3.5.8)	List of scripts (page 357) Upgrading from 3.4.4 to 3.5.0 (page 311) Upgrading from 3.5.x to 3.5.y (page 307)
from 3.5 to 3.6.5	convertxmllinks.php updaterelatedobjectlinks.php updateeztimetype.php (if not previously run when upgrading to 3.5.3+) updatecrc32.php (if not previously run when upgrading to 3.5.8+)	List of scripts (page 357) Upgrading from 3.5.2 to 3.6.0 (page 299)
from 3.6 to 3.8.10	updatemultilingual.php updaterssimport.php (optional) correctxmltextclasses.php (added in 3.8.1) updatetipafriendpolicy.php (added in 3.8.9) updatecrc32.php (if not previously run when upgrading to 3.5.8+ or 3.6.5+)	List of scripts (page 357) Upgrading from 3.6.x or 3.7.x to 3.8.0 Upgrading from 3.8.x to 3.8.y (page 280)

How to run the system upgrade scripts

All the scripts must be run from the root directory of your eZ Publish installation. Read the information indicated in the list of the system upgrade scripts (page 357) for the script before running it. Note that sometimes a specific configuration change needs to be done before running a particular upgrade script. For more detailed information about how to run each script, refer to the third step in the upgrading instructions for the version of eZ Publish in which this script was added. You can also find a few general recommendations below.

In most cases, the only thing that an upgrade script does is modify some data in the database. Such scripts should be run once for each database, specifying one siteaccess per database. If you only have a public and an administration siteaccess that share the same database (which is the most typical/usual case), you only need to run the script for one of the siteaccesses.

1. Navigate into the eZ Publish 3.x.y directory.
2. Run the script (replace <script> with the actual name of the script and <siteaccess> with the name of your siteaccess(es)):

```
php update/common/scripts/<script>.php -s<siteaccess>
```

This requires that you have [PHP CLI](#) enabled. Note that sometimes it is also possible (but not recommended since it might cause problems) to use the CGI version, which supports the CLI behavior by means of the -C switch when run from the command line:

```
php -C update/common/scripts/<script>.php -s<siteaccess>
```

In eZ Publish 3.0-2 and later versions, the system upgrade scripts are executable so that they can be run as normal shell scripts on a Linux/UNIX based system:

```
update/common/scripts/<script>.php -s<siteaccess>
```

Use "--help" or "-h" to output the information about the script and its parameters:

```
php update/common/scripts/<script>.php --help
php update/common/scripts/<script>.php -h
```

Step 4: Updating the system configuration

After running the upgrade scripts above you may need to review some of your configuration settings in order to be compatible with the changes that have been added. Read the fourth step of the upgrading instructions for each particular version of eZ Publish that is between 3.a.b and 3.x.y.

For example, when upgrading directly from 3.5.11 to 3.6.11, you should read the fourth step of the instructions in the "Upgrading from 3.5.2 to 3.6.0 (page 299)" and "Upgrading from 3.6.x to 3.6.y (page 296)" manuals. According to these instructions, you may need to do the following:

- Enable new toolbars in the administration interface.
- Configure the system to show hidden nodes in the administration interface.
- Create custom classes for the newly added "embed" XML tag, and add a new template for rendering the content of this tag.
- Enable database transaction support.
- Review configuration settings for the "literal" XML tag.

Step 5: Clearing the caches

Whenever an eZ Publish solution is upgraded, all caches must be cleared in a proper way. This should be done from within a system shell:

1. Navigate into the eZ Publish 3.x.y directory.
2. Run the clear cache script:

```
bin/shell/clearcache.sh --clear-all
```

If you have [PHP CLI](#) installed, you can also do the following (this script should also work on Windows):

```
php bin/php/ezcache.php --clear-all --purge
```

From 3.10, the "clearcache.sh" script has been disabled. This means that the file is still present but the code inside is removed. It outputs a message telling the user to use "bin/php/ezcache.php". In other words, you have to use "ezcache.php" instead.

Sometimes the script is unable to clear caches because of restrictive file/directory permission settings. Make sure that all caches have been cleared by inspecting the contents of the various cache subdirectories within the "var" directory (typically the "var/cache/" and "var/<name_of_siteaccess>/cache/" directories). If there are caches left, you need to remove them manually.

1.27 The system upgrade scripts

This section reveals the list of the system upgrade scripts located either in the "update/common/scripts" or "bin/php" directory. The scripts are included in various eZ Publish 3 releases and intended to update the existing site data when upgrading the system from older versions. Please refer to the "Upgrading from 3.a.b to 3.x.y" section to learn how to figure out which scripts that should be run in each particular case. (Note that some of the scripts listed below, e.g. "update-searchindex.php" and "updateniceurls.php" can be used at all times; not just during upgrades.)

The main scripts for system upgrading (v3.10)

Version	Script name	Description
3.10.0	updateniceurls.php	In eZ Publish 3.10, a new feature that makes it possible to use multi-lingual virtual URLs has been added. In order to use this functionality, you will have to set the desired type of URL transformation and run the "updateniceurls.php" script. It will go through all content objects stored in the database and create new virtual URLs for them in accordance with the selected type of transformation. Refer to the upgrade instructions for more information.
3.10.0	ezimportdbfile.php updateisbn13.php	In eZ Publish 3.10, the different ranges are stored in the database, allowing the system to fix up incorrectly entered ISBN-13 codes. When upgrading, you need to import ranges to the database manually. This must be done even though your site does not use ISBN-13 numbers. You can either import ranges from the "db_data.db" file located in the "kernel/classes/ezisbn/share/" directory using the "ezimportdbfile.php" script or down-

		load the latest "range.js" file and run the "updateisbn13.php" script. Refer to the upgrade instructions for more information.
3.10.0	ezconvert2isbn13.php	In order to use ISBN-13 numbers after upgrading to eZ Publish 3.10, you need to run the "ezconvert2isbn13.php" script. It will go through all content object attributes using the "ISBN" datatype and convert their values to valid ISBN-13 numbers. Refer to the upgrade instructions for more information.
3.10.0	updatemultioption.php	In eZ Publish 3.10, a new datatype called "Multi-option2" has been added as a replacement for the old "Multi-option" datatype. In order to be compatible with this datatype, you have to run the "updatemultioption.php" script. The script will go through all content object attributes that are using the old "Multi-option" datatype and make sure (by the way of conversion) that they are using the new "Multi-option2" datatype. Refer to the upgrade instructions for more information.

The main scripts for system upgrading (v3.9)

Version	Script name	Description
3.9.3	updatevatcountries.php	As a result of an important bugfix related to the handling of country dependent VATs, changes have been made in eZ Publish 3.9.3 (and in version 3.10.0 as

		well). If you are upgrading from version 3.9.2 or earlier, you need to run the "updatevatcountries.php" script. The script will go through all the VAT rules in the database and make them compatible with the introduced changes. Refer to the upgrade instructions for more information.
3.9.0	updateclasstranslations.php	In eZ Publish 3.9, a new feature has been added to the multi-language functionality that makes it possible to translate names of the eZ Publish content classes / class attributes. In order to make sure that your site is compatible with this functionality, you will have to run the "updateclasstranslations.php" script. This step must be done even though your site has only one language. The script will check your content classes and content class attributes and re-initialize their names for missing translations. Refer to the upgrade instructions for more information.
3.9.0	correctxmltext.php	eZ Publish 3.9.0 supports using custom attributes in all XML tags. The names of the custom attributes must be added to the global (or siteaccess) "content.ini.append.php" configuration file. In order to be compatible with this feature, you will have to run the "correctxmltext.php" script. The script will go through the XML block attributes stored in the database, convert the deprecated "object" tags to

		<p>”embed” tags, add missing settings for all custom classes and custom attributes used within XML tags. Note that this script has been improved in eZ Publish 3.9.1. If you are upgrading to 3.9.0, it is recommended to copy the script from 3.9.1. Refer to the upgrade instructions for more information.</p>
3.9.0	updatetypedrelation.php	<p>In eZ Publish 3.9.0, the object relations functionality has been improved. In order to be compatible with this feature, you will have to run the ”updatetypedrelation.php” script. The script will go through the XML block attributes stored in the database and create object relations of the ”XML linked” and ”XML embedded” types if needed. Refer to the upgrade instructions for more information.</p>

The main scripts for system upgrading (v3.8)

Version	Script name	Description
3.8.9	updatetipafriendpolicy.php	<p>In order to make it possible to restrict access to the ”Tip a friend” feature, changes have been made in eZ Publish 3.8.9 (and to versions 3.9.3 and 3.10.0 as well). Users (typically the anonymous role) need to have access to the ”tipafriend” function of the ”content” module in order to use the ”Tip a friend” feature. You need to run the ”updatetipafriendpolicy.php” script. The script will add a new role that</p>

		grants access to the "Tip a friend" feature and ask for confirmation about assigning it to each user or user group located one level beneath the "Users" top level node. Refer to the upgrade instructions for more information.
3.8.1	correctxmltextclasses.php	In eZ Publish 3.8.0 and earlier versions, you could use custom classes in XML tags without having to specify them first. From 3.8.1, you have to specify the names of all custom classes that are used within XML tags. It is recommended (but not required) to run the "correctxmltextclasses.php" script when upgrading to 3.8.1 or later version. The script will go through the XML block attributes stored in the database, check which classes are in use and add missing settings (if any) to the "content.ini.append.php" configuration file. Refer to the upgrade instructions for more information.
3.8.0	updatemultilingual.php	The 3.8.0 version of eZ Publish introduces improved multi-language functionality. In order to make sure that your site is compatible with this feature, you will have to run the "updatemultilingual.php" script. This step must be done even though your site has only one language. Note that this script has been improved in eZ Publish 3.8.1. If you are upgrading to 3.8.0, it is recommended to copy the script from 3.8.1. Refer to the up-

		grade instructions for more information.
3.8.0	updaterssimport.php	The 3.8.0 version of eZ Publish introduces more generic RSS import that allows using custom RSS import definitions. If you are using the RSS import functionality on your site, you will have to run the "updaterssimport.php" script. Refer to the upgrade instructions for more information.

The main scripts for system upgrading (unsupported versions)

Version	Script name	Description
3.0-1	updatesearchindex.php	In eZ Publish 3.0-1, the built-in search engine has been improved (see the 3.0-1 changelog for more information) and thus you need to update the search index of your site when upgrading from 2.9-x. To do this, clear the search index tables in your database (these are called "ezsearch_word" and "ezsearch_object_word_link") and run the "updatesearchindex.php" script. The script must be run only once, not per siteaccess.
3.0-1	updateniceurls.php	In eZ Publish 3, the virtual URLs (nice URLs, URL aliases) are automatically generated and maintained for all nodes. The generated URL for a node is based on the node's location in the tree and the actual name of the object that the node encapsulates. If you are upgrading from 2.9.x then you should

		<p>run the "updateniceurls.php" script in order to generate virtual URLs for all your content nodes. This script only modifies the "ezurlalias" table in the database. It must be run only once, not per siteaccess.</p> <p>Please note that this script has been improved in eZ Publish 3.2-2. If you are upgrading to a version prior to 3.2-2, it is recommended to copy the script from 3.2.-2.</p>
3.0-1	updatexmltext.php	<p>In eZ Publish 3.0-1, the "ezxmltext" datatype has been changed in order to store an XML code using UTF-8 encoding (see the 3.0-1 changelog for more information). If you are upgrading from 2.9.x then you should run the "updatexmltext.php" script for automatic conversion of your XML fields to new format. It is recommended to create a database backup before using this script. The script must be run only once, not per siteaccess.</p>
3.1.0 rel 1	generateoverridesettings.php	<p>In eZ Publish 3.1.0 release 1, the new template override subsystem that allows to use custom templates instead of the default/standard ones has been implemented (see the 3.1.0 rel1 changelog for more information). The template overrides must be defined in the "override.ini.append.php" file of a siteaccess.</p> <p>If you are upgrading to 3.1.0 rel1 or later then you should run the "genera-</p>

		<p>teoverridesettings.php” script for each of your siteaccesses. The script will output a suggestion for the siteaccess ”override.ini.append.php” file that needs to be created for the specified siteaccess. You should create a new file called ”override.ini.append.php” in the ”settings/siteaccess/example” directory (replace ”example” by the actual name of your siteaccess) and copy the output provided by the script to this file.</p> <p>Please note that this script has been improved in eZ Publish 3.2-1 (3.2.0 release 3). If you are upgrading to a version prior to 3.2-1, it is recommended to copy the script from 3.2-1.</p>
3.2-1	updateattributefilter.php	<p>eZ Publish 3.2-1 supports sorting and filtering on attribute level. It is possible to sort and/or filter a list of content objects based on the values of a specific attribute (see the 3.2-1 changelog for more information). The ”list” and ”tree” fetch functions support this functionality.</p>
3.2-4	updatetranslations.php	<p>In eZ Publish 3.2-4, an important bug related to adding new object attributes has been fixed (see the 3.2-4 changelog for more information). Since that bug might lead to missing translated attributes on multilingual sites, it is recommended to run the ”updatetranslations.php” script. The</p>

		script will check your content objects and re-initialize attributes if their data is missing as a result of the bug.
3.4.0	updateremoteid.php	<p>In eZ Publish 3.4.0, a new attribute called "remote_id" has been added to content objects, classes and nodes. This attribute stores a global unique ID for an object/class/node. The remote ID is an MD5 hash of the time when the object/class/node was created plus a random value. These IDs are used in order to avoid collisions of identical objects/classes/nodes when importing and exporting data from one site to another.</p> <p>If you are upgrading to 3.4.0 then you should run the "updateremoteid.php" script in order to make sure that all your items that support the remote ID gets one generated. The script must be run only once, not per siteaccess.</p>
3.4.0	updatesession.php	<p>The eZ Publish session support is based on the PHP session functionality. Most of the session handling work is done by PHP; some eZ Publish specific functions are called for getting and storing the session data in the database. In eZ Publish 3.4.0, this functionality has been improved in order to relate a session to a user (see the 3.4.0 changelog for more information). The session data in the "ezsession" table now contain information about the session user. This allows to easily count</p>

		<p>the number of active registered users and anonymous users. When a user is being removed or disabled, the system will remove the corresponding session(s) in order to automatically log out invalid users.</p> <p>If you are upgrading from 3.3-x to 3.4.0 then you should run the "updatesession.php" script in order to update the entries in the "ezsession" table. The script must be run only once, not per siteaccess.</p>
3.4.0	updatetoplevel.php	<p>In eZ Publish versions prior to 3.4, there were three top level nodes ("Content", "Media" and "Users"). Since in eZ Publish 3.4.0 a new one called "Setup" has been added, it is recommended to run the "updatetoplevel.php" script when upgrading. The script will check the top level nodes and match them with the values defined under the "[NodeSettings]" section of the "content.ini" configuration file or its override. If "content.ini" or its override contains incorrect IDs of the top level nodes, these will be automatically updated. If some of the top level nodes are missing, these will be created. The script must be run only once, not per siteaccess.</p>
3.4.0	addorderemail.php	<p>In eZ Publish 3.4.0, the webshop functionality has been improved (see the 3.4.0 changelog for more information). The list of orders that is stored in the "ezorder" database table now includes</p>

		<p>the customer's email for each order. If you are using the eZ Publish webshop functionality on your site and your order list is not empty, then you need to run the "addordere-mail.php" script when upgrading to 3.4.0. The script will modify the "ezorder" table in the database. It must be run only once, not per siteaccess.</p>
3.5.0	updateiscontainer.php	<p>In eZ Publish 3.5.0, a new attribute called "is_container" has been added to content classes (see the feature doc for more information). The container flag controls whether an instance of the class should be allowed to have sub items (often called child nodes, children) or not. If you are upgrading to 3.5.0 then you should run the "updateiscontainer.php" script in order to set this flag for those classes that are allowed to have subitems. The script must be run for each of your siteaccesses. Please note that the script can only modify the general purpose classes. You should manually set the container flag for your custom classes (if there are any), otherwise it won't be possible to create subitems.</p>
3.5.0	updateniceurls.php	<p>eZ Publish 3.5 includes the built-in transliteration feature that allows to automatically convert special characters to their equivalent in English letters when generating the virtual URLs (nice URLs, URL aliases). For example, the Norwegian characters " ",</p>

		<p>” and ”” are converted to ”ae”, ”oe” and ”aa” instead of underscores.</p> <p>If you are upgrading to 3.5.0 or later then it is recommended (but not required) to run the ”updateniceurls.php” script in order to make sure all your content nodes have their virtual URLs generated using a correct transformation of special characters. The script modifies the ”ezurlalias” table in the database. It must be run only once, not per siteaccess.</p>
3.5.0	updatesearchindex.php	<p>eZ Publish 3.5.0 includes a few bug fixes related to the search index generation (see the 3.5.0 changelog for more information). It is recommended (but not required) to update the search index of your site when upgrading to 3.5.0. This can be done by running the ”update-searchindex.php” script with the ”--clean” option. This option (added in eZ Publish 3.3-1) means that the existing search index data should be removed before starting indexing. The script must be run only once, not per siteaccess.</p>
3.5.1	updatetoplevel.php	<p>eZ Publish versions prior to 3.5.1 suppose that there are four top level nodes (”Content”, ”Media”, ”Users” and ”Setup”). In eZ Publish 3.5.1 a new one called ”Design” has been added.</p> <p>If you are upgrading to 3.5.1 then you should run the ”updatetoplevel.php” script for your admin siteaccess. The</p>

		<p>script will check the top level nodes and match them with the values defined in the "menu.ini" configuration file or its override. (From 3.5.0, the main menu in the admin interface can be configured using "menu.ini".) If "menu.ini" or its override contains incorrect information about the top level nodes, it will be automatically updated. If some of the top level nodes are missing, these will be created.</p> <p>Please note that this script has been improved in eZ Publish 3.5.2. If you are upgrading to a version prior to 3.5.2, it is recommended to copy the script from 3.5.2.</p>
3.5.3	updateeztimetype.php	<p>In eZ Publish 3.5.3, the "Time" datatype has been modified in order to make it possible to interpret the value of hours and minutes properly and independently from DST and GMT transformations (see the feature doc for more information).</p> <p>If you are upgrading to 3.5.3 or later, then you should run the "updateeztimetype.php" script in order to convert the attributes of the "eztime" datatype from GMT to the server local time. You need to create a database backup before using this script. The script must be run for each siteaccess.</p>
3.5.8	updatecrc32.php	<p>Since some PHP functions (like crc32 and ip2long) work in different ways on 64bit and 32bit servers, changes has been</p>

		made in eZ Publish 3.5.8 (and to versions 3.6.5, 3.7.3 and 3.8.0 as well) in order to avoid problems when migrating from 32bit to 64bit architecture. You need to run the "updatecrc32.php" script when upgrading even if you don't have a 64bit machine. (The script will update old signed to unsigned values in the database.) It is also recommended to specify "64bit-CompatibilityMode=enabled" in your "site.ini.append.php" configuration file.
3.6.0	convertxmllinks.php	In eZ Publish 3.6, new XML tags called "link" and "embed" that replace and extend functionality of the old "object" tag have been introduced (see this page for more information). In order to be compatible with this feature, you will have to run the "convertxmllinks.php" script. The script will update the tags in the attributes of the "ezxml-text" datatype. It should be run for all siteaccesses that use different databases. If you only have a public and an administration siteaccess (which is the most typical/usual case), then you only need to run the script for one of the siteaccesses. If the siteaccess isn't specified, the default siteaccess will be used.
3.6.0	updaterelatedobjectlinks.php	In eZ Publish 3.6, the related objects functionality has been improved (see this page for more information). In order to be compatible

with this feature, you will have to run the "update-relatedobjectlinks.php" script. This script should be run for all siteaccesses that use different databases. If you only have a public and an administration siteaccess (which is the most typical/usual case), then you only need to run the script for one of the siteaccesses.

The additional scripts (tools for data updating)

Version	Script name	Description
3.3-1	cleanup.php	eZ Publish 3.3-1 includes a new script called "cleanup.php". It allows to clean up the database in order to get rid of various data that should not be included in release (see the 3.3-1 changelog for more information). The script can be used to remove outdated sessions, information related to user preferences, workflows, collaboration system, notifications etc.
3.3-1	flatten.php	The "flatten.php" script has been added in eZ Publish 3.3-1 and moved from the "update/common/scripts" directory to "bin/php" in 3.5 (rev. 13488). This script allows to remove the data that is not considered currently in use from the database. This script will remove any non-published version, leaving only the one with the PUBLISHED status. In this case, you will probably need to run the "flatten.php" script

		<p>in order to delete unneeded versions from the previously published objects. This script is often used to clean the database at the end of the development process before launching the site.</p>
3.5.0	updatecontentobjectname.php	<p>In eZ Publish versions prior to 3.5, updating the object name pattern of a class affected only new objects of this class. To update the names of the existing objects of this class accordingly, you had to re-publish each of them. In eZ Publish 3.5, a new script called "updatecontentobjectname.php" has been added for automatic update of the names of the content objects. After changing the object name pattern for a class, you can update the names of the existing content objects by running this script for your siteaccess. It is recommended to create a database backup before using this script. You may also need to update your nice URLs and search index using the "updateniceurls.php" and "updatesearchindex.php" scripts.</p>

1.28 Notes about database changes

This section explains which database upgrade scripts can be skipped when upgrading an eZ Publish database to a newer version. In addition, it provides an overview of the database schema changes within stable release branches of eZ Publish.

Version-focused database upgrade scripts

When upgrading an eZ Publish database from version X to version Y, you have to run a set of the database upgrade scripts following the official upgrade path. This basically means that you need to run one database upgrade script for each stable release between X and Y. However, it often happens that a ".sql" file of the script only contains one or two lines with something like this:

```
UPDATE ezsite_data SET value='Z' WHERE name='ezpublish-version';
UPDATE ezsite_data SET value='N' WHERE name='ezpublish-release';
```

In this case, a database upgrade script is called "version-focused", since it does not do anything except update the current version number, which is stored internally in the system. When upgrading an eZ Publish database, you can skip all version-focused scripts for intermediary versions, as shown in the example below.

Example

In order to upgrade an eZ Publish database from version 3.6.0 to 3.8.10, you should run eleven database upgrade scripts:

- dbupdate-3.6.0-to-3.8.0.sql
- dbupdate-3.8.0-to-3.8.1.sql
- dbupdate-3.8.1-to-3.8.2.sql
- dbupdate-3.8.2-to-3.8.3.sql
- ...
- dbupdate-3.8.9-to-3.8.10.sql

There are seven version-focused upgrade scripts, meaning that seven of the scripts in the list contain only two lines (that simply update the version number) and can be skipped. Therefore, the list of scripts to run contains four items instead of eleven:

- dbupdate-3.6.0-to-3.8.0.sql
- dbupdate-3.8.0-to-3.8.1.sql

- dbupdate-3.8.4-to-3.8.5.sql
- dbupdate-3.8.9-to-3.8.10.sql

Running these four scripts is equivalent to running all the eleven scripts mentioned above.

Database schema changes

If a database upgrade script is not version-focused (contains more than two lines), this usually means that the database structure was changed since the previous release. This is very typical for the first stable release in a branch, since there are usually some database schema changes as compared to the previous branch.

Within stable release branches, database schemas are generally not changed. In other words, once the first stable release in a branch is out, the database schema should remain unchanged. However, exceptions are made for changes that are needed for important bugs and enhancements. When the next stable release in the current branch is out, the change will be present in the database upgrade script for it. In addition, such a bug fix also gets included in the next / upcoming branch for which there are no stable releases yet; the fix will be present in the database upgrade script for the first stable release in this branch (these are mentioned as 3.x.0 versions in the table below).

The following table provides an overview of the database changes within stable release branches of eZ Publish, starting from version 3.6.

Database changes	Introduced in
UPDATE ezpolicy SET function_name='administrate' WHERE module_name='shop' AND function_name='adminstrate';	3.6.1, 3.5.4, 3.7.0, 3.8.0
Bug fix for problems with performance of the "Keywords" datatype and missing indexes in database tables .	3.6.9, 3.7.7, 3.8.1, 3.9.0
Bug fix for missing indexes in 3.8 database .	3.8.5, 3.9.0
Bug fix for problem with storing long URLs .	3.8.10, 3.9.4, 3.10.0
	3.9.1, 3.10.0

<pre>ALTER TABLE ezcontentclass CHANGE COLUMN serialized_name_list serialized_name_list longtext default NULL; ALTER TABLE ezcontentclass_attribute CHANGE COLUMN serialized_name_list serialized_name_list longtext NOT NULL;</pre>	
<pre>ALTER TABLE ezvatrule CHANGE country country_code varchar(255) DEFAULT '' NOT NULL;</pre>	3.9.3, 3.10.0

Shyare

information