



Escenic Content Engine  
**Migration Guide**  
5.2.7.2







Copyright © 2009-2011 Vizrt. All rights reserved.

No part of this software, documentation or publication may be reproduced, transcribed, stored in a retrieval system, translated into any language, computer language, or transmitted in any form or by any means, electronically, mechanically, magnetically, optically, chemically, photocopied, manually, or otherwise, without prior written permission from Vizrt.

Vizrt specifically retains title to all Vizrt software. This software is supplied under a license agreement and may only be installed, used or copied in accordance to that agreement.

### **Disclaimer**

Vizrt provides this publication “as is” without warranty of any kind, either expressed or implied.

This publication may contain technical inaccuracies or typographical errors. While every precaution has been taken in the preparation of this document to ensure that it contains accurate and up-to-date information, the publisher and author assume no responsibility for errors or omissions. Nor is any liability assumed for damages resulting from the use of the information contained in this document.

Vizrt’s policy is one of continual development, so the content of this document is periodically subject to be modified without notice. These changes will be incorporated in new editions of the publication. Vizrt may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time.

Vizrt may have patents or pending patent applications covering subject matters in this document. The furnishing of this document does not give you any license to these patents.

### **Technical Support**

For technical support and the latest news of upgrades, documentation, and related products, visit the Vizrt web site at [www.vizrt.com](http://www.vizrt.com).

### **Last Updated**

21.12.2011





# Table of Contents

<b><a href="#">1 Introduction</a></b> .....	<b>7</b>
<b><a href="#">1.1 Differences</a></b> .....	<b>7</b>
<b><a href="#">2 The Migration Procedure</a></b> .....	<b>11</b>
<b><a href="#">2.1 Preparation</a></b> .....	<b>11</b>
<b><a href="#">2.2 Test Migration</a></b> .....	<b>11</b>
<b><a href="#">2.3 Problem Solving</a></b> .....	<b>14</b>
<b><a href="#">2.4 Scaling Up</a></b> .....	<b>14</b>
<b><a href="#">2.5 Full-Scale Migration</a></b> .....	<b>15</b>
<b><a href="#">3 Import</a></b> .....	<b>17</b>
<b><a href="#">3.1 Migration Import Filters</a></b> .....	<b>17</b>
<b><a href="#">3.2 Splitting Large Import Files</a></b> .....	<b>18</b>
<b><a href="#">3.3 HTML Markup</a></b> .....	<b>19</b>
<b><a href="#">3.3.1 Configuring the HTML Clean-up Process</a></b> .....	<b>20</b>
<b><a href="#">3.4 Avoiding Duplicate URIs</a></b> .....	<b>22</b>
<b><a href="#">4 Upgrading Publication Web Applications</a></b> .....	<b>23</b>
<b><a href="#">4.1 Convert Publication Resources</a></b> .....	<b>23</b>
<b><a href="#">4.2 Handle Relation Types</a></b> .....	<b>25</b>
<b><a href="#">4.3 Upgrade WEB.XML</a></b> .....	<b>25</b>
<b><a href="#">4.4 Upgrade Templates</a></b> .....	<b>26</b>
<b><a href="#">4.5 Upgrade Java Code</a></b> .....	<b>26</b>





---

# 1 Introduction

This manual is intended as an aid to system administrators responsible for the migration of Escenic publications from version 4.x of the Content Engine to version 5.2.7.2. It is intended to provide assistance both during the planning and execution phases of a migration.

In outline, the migration of a publication from 4.x to 5.2.7.2 involves the following tasks:

1. Export the entire publication from the old Content Engine.
2. Import all the exported syndication files to the new Content Engine.
3. Upgrade the publication templates.

Vizrt has invested considerable effort in making this migration process as simple and robust as possible, but it is neither entirely automatic nor error-proof, and should be approached with care. The method described in this manual involves:

- Starting with a small sample taken from one publication, and testing with that until an error-free migration is achieved
- Then scaling up to larger samples until you are confident that you have a working migration procedure

For any large installation, migration is a time-consuming process, so the migration tools provided by Vizrt facilitate running the old and new systems in parallel. It is possible, for example, to continue editorial work on the old system for a while, but switch to the new system for publication purposes. The migration tools can be set up to perform continuous migration of new and changed content between the two systems, so that the publication system shadows the editorial system with just a few minutes delay.

As mentioned in [section 1.1](#), one of the new features in version 5 is a new and more flexible structure for section pages. If you have been dissatisfied with the front page layout limitations in version 4, then you may wish to carry out a redesign of your section pages in combination with the upgrade. For further information, see [section 4.1](#).

---

## 1.1 Differences

Version 5 is a major upgrade of the Content Engine with significant changes to both the user interface and the underlying architecture. The major differences are:

### **New Content Studio**

Content Studio has been completely re-written, and now has a much more modern and user-friendly interface, with a new rich text editor for editing HTML fields. The new Content Studio is the only means of editing

content in version 5. For details, see the **Escenic Content Studio User Guide**.

### **Smaller Web Studio**

The scope of Web Studio has been reduced in version 5; it is now a publication management tool, and all content editing functionality has been removed. Publication administrators can use Web Studio for general publication management, editing the section structure of publication and access rights administration. For details, see the **Escenic Content Engine Publication Administrator Guide**.

### **Everything is content**

The image, multimedia and link objects used to represent binary data in version 4 no longer exist, and articles have been renamed: they are now called **content items**. This name change reflects the fact that content items are now used to represent all kinds of content, including binary content such as images, audio, video, and binary documents such as PDFs, Word files, spreadsheets and so on. However, template developers and programmers should note that for reasons of backwards compatibility, "article" has been retained in tag, class and method names. In addition, a number of transitional implementations of old image/multimedia/link classes and tags have been made in order to ensure backwards compatibility for version 4 templates. For a detailed description of the new architecture, see the **Escenic Content Engine Template Developer Guide**.

### **Catalogs withdrawn**

Catalogs do not exist in version 5. Since images and other multimedia objects are now represented by standard content items, there is no longer any need for a special means of organizing them. They can be organized using a publication's section structure in the same way as all other content.

### **Section pages**

Section front pages are now simply called **section pages**. A section can have any number of section pages, and section pages have a new and more flexible internal structure based on new concepts called **groups** and **areas**.

### **Image versions**

A new, more flexible concept for managing multiple versions of images, called **representations**, has been introduced. The old image version concept is, however, retained for reasons of backwards compatibility.

### **New resource file formats**

New XML resource file formats supporting the new concepts and structures are introduced. In addition, the `articleType` resource has been renamed to `content-type`. For details, see the **Escenic Content Engine Resource Reference**.

**New syndication format**

The version 4 import/export format has been replaced by a new **syndication format** that supports all version 5 concepts and structures. For details, see the **Escenic Content Engine Syndication Reference**. In order to support migration, version 5 is still capable of importing version 4 import/export format as long as version 4-specific structures are avoided. The migration tools described in this manual include special import filters that convert such version 4-specific to their version 5 equivalents during import. For more information about this, see [chapter 3](#).

**XML only**

HTML fields in version 4 articles were very permissive, and would accept a wide range of HTML variants. Version 5 content items, on the other hand may only contain well-formed XML (XHTML plus Escenic `relation` elements). In order to support migration, the version 5 import process checks imported markup and automatically corrects it where possible. For further information, see [section 3.3](#).

---

In general, version 5 terminology is used throughout this manual. Version 4 terms are only used when referring specifically to version 4 concepts.

---



---

## 2 The Migration Procedure

This chapter contains an outline description of a migration procedure. It is intended as a guide to help ensure that you think of all the issues that you will need to take care of, and to indicate the general order in which tasks should be carried out. It is not necessarily the best procedure for all installations, and you may well need to modify it to meet your specific requirements.

---

### 2.1 Preparation

You should not attempt to migrate any of your publications before you have:

- A functioning Content Engine 5.2.7.2 installation to which publications can be migrated.
- Sufficient staff (both technical and editorial) who are familiar with the new version of the software and are capable of maintaining the migrated publication.
- A plan for carrying out the migration.
- A 4.x host available for export purposes. This host must **not** be a host that you use for editorial or presentation purposes, as you will need to reconfigure the search engine in a way that will degrade search performance for users.
- Sufficient spare disk capacity for handling the export/import data. Copies of all images and multimedia files are created during export, and this will most likely account for the majority of your space requirements. In addition, you will need some additional space for the XML syndication files generated during export, but this is likely to form a relatively small proportion of the total. The syndication files exported from our test database consisting of around 300,000 articles require 6.5Gb of space, but the space required for associated images and multimedia files is far greater. It is probably a good idea to mount a network storage device with sufficient storage on both your import and export hosts: this will also simplify transferring data between the machines.

---

If the server runs out of disk space during export, the export process is paused.

---

Once these prerequisites are met, you should install the Migrator plug-in on the version 4.x host you have designated as your export host. For detailed instructions on how to install and use this plug-in, see the **Escenic Content Engine Migrator Plug-in Guide**.

---

### 2.2 Test Migration

Once your preparations are complete, you should carry out a test migration of part of a publication. You should try to choose a publication that has a wide

range of content types in order to ensure as many variations as possible are tested.

Before you start the test migration you need to decide how much of the publication you are going to use for test purposes. You should use enough to ensure good coverage of all the types of content in your publications, but small enough that the migration process doesn't take too long (you may need to repeat it several times). You should aim for a few thousand articles (probably somewhere between a week and a month of content).

The general migration procedure is as follows:

1. Create a version 5.2.7.2 copy of the publication you have chosen. This involves converting your old publication resources to the new version 5.2.7.2 formats. For details of how to do this, see [section 4.1](#). For general information on version 5.2.7.2 publication resources, see the **Escenic Content Engine Resource Reference**. For detailed instructions on how to create a publication in version 5.2.7.2, see **Escenic Content Engine Server Administration Guide, section 2.1.10**.

You must give your new publication the same name as your old one.

You should also make sure that the new publication you have created is empty, since any content in the publication may conflict with the content migrated from version 4. If the publication archive (WAR file) from which you create the new publication contains a `/META-INF/escenic/resources/escenic/content` file, its contents are automatically imported as part of the publication creation process. This means that the new publication will have some default content as soon as it is created. In order to avoid this happening you should either:

- Click on the file's **Remove uploaded resource** link before creating the publication.
  - Remove this file from the publication archive before you upload it.
2. Configure the Migrator plug-in to export the "one-time" items by setting the following parameters in the **Migration configuration tool**:
    - **Publications** (set to the name of your chosen publication)
    - **One time exports** (set to `yes`)

Then click on **Reconfigure** to apply the changes. The Migrator will then start exporting the selected publication's "one-time" items (section structure, users, user groups, access control lists and so on) to the export folder shown in the configuration tool.

3. Copy the exported files to the publication's import folder on your version 5.2.7.2 host, and import them. (See [section 3.1](#) for the location of this folder.) For details of how to import content to version 5.2.7.2, see the **Escenic Content Engine Syndication Reference**. It is important to import the exported data in the correct order, which is:

users  
user groups



sections  
access control lists

This ensures that all referenced objects already exist in the database and that the required links can be created.

4. Use Web Studio to verify that the data has been correctly imported: that the new publication has the expected users, section tree and access rights. For information on how to use Web Studio, see the **Escenic Content Engine Publication Administrator Guide**.
5. Reconfigure the Migrator to export all articles for the period you have selected by setting the following parameters in the **Migration configuration tool**:
  - **Publications** (set to the name of your chosen publication)
  - **Content Type** (select all the article types you want to export)
  - **One time exports** (set to **No**)
  - **Start Date** (set, for example, to one month ago)
  - **Interval** (Setting this to 30 days, for example, will mean that the whole period is exported in a single pass, with one output file for each content type. Setting it to 1 day will result in up to 30 output files for each content type and will take at least about half an hour, since the export passes are started at 1 minute intervals.)
  - **Reference Types** (Check all boxes)

Then click on **Reconfigure** to apply the changes. The Migrator will then start exporting data to the export folder shown in the configuration tool. How long the entire process takes will depend on how much content you have chosen to export.

6. Any images in the articles you have exported will result in one or more export files in an **images** sub-folder - for example

```
/path-to-export-folder/publication-name/images/images_from_start-date_to_end-date.xml
```

Along with these files will be all the image files they reference. Copy all the file(s) in the **images** folder (both the XML files and image files) to your import folder and import them.

7. Any multimedia objects in the articles you have exported will result in one or more export files in a **media** sub-folder - for example

```
/path-to-export-folder/publication-name/media/media_from_start-date_to_end-date.xml
```

Along with these files will be all the multimedia files they reference. Copy all the file(s) in the **media** folder (both the XML files and media files) to your import folder and import them.

8. When the multimedia import is completed (the import folder is empty), copy all the remaining standard article files to the import folder and import them.
9. When the main import is completed (the import folder is empty), copy all the standard article files to the import folder again, and **import them**

**a second time.** This is necessary to ensure that all relations between content items are created. A relation can only be created if both related objects are available, and the only way to ensure that this is the case is to import everything twice.

10. Log into the new publication using Content Studio 5.2.7.2 and examine it carefully, making sure that you find everything you expect to find:
  - Images and multimedia objects open and are correct
  - URLs are the same
  - Relations have been correctly established

Note that the Preview function will not yet work.

11. Make a copy of your old publication web application, and apply the changes described in [chapter 4](#) to it.
12. Deploy the upgraded web application on version 5.2.7.2 (see the **Escenic Content Engine Template Developer Guide** for more information about this).
13. Start up a browser and try to view the new publication.

---

## 2.3 Problem Solving

Most likely you will encounter a few problems with your test migration. The general procedure for dealing with any problems you find is:

1. Locate the problem.
2. Make whatever changes you think are necessary to fix it.
3. Delete the new publication.
4. Repeat the test migration.

This sequence should be repeated until the results of the test migration are satisfactory.

You won't want to repeat the whole test migration for every single problem you find. You may find that you can provoke a number of problems by migrating a much smaller data sample, for example, or all your problems may be import-related, so you can investigate them by repeatedly importing a single file.

Both the export and import processes report their progress and/or failures by writing messages to the Content Engine log, so this is a valuable source of diagnostic information. For instructions on how to configure Content Engine 5.2.7.2 log output, see **Escenic Content Engine Server Administration Guide, chapter 11**.

If you encounter problems that you cannot fix, please contact Vizrt support.

---

## 2.4 Scaling Up

Once your test migration works satisfactorily, it is time to start the process of scaling up to full-scale migration. Precisely how you do this will depend very



much on your particular circumstances, and also to some degree on how many problems you encountered with your initial test migration. Some alternatives are:

- Repeat the test migration on the same publication, but with a larger data sample
- Progress immediately to migrating the entire test publication
- Repeat the test migration with one or more different publications that have very different content types

---

## 2.5 Full-Scale Migration

At some point you will have gained sufficient confidence in the results of your tests to start migrating "for real".

The procedure for a "production" migration is basically the same as the test procedure described in [section 2.2](#). Here is a list of issues to be aware of when preparing for production migration:

### Disk capacity

The data volumes involved are going to be much larger when you scale up. Make sure you have enough spare disk capacity on both your export and import machines.

### Export system cache sizes

Exporting all the content from a publication has the side effect of loading all that content into the Content Engine's caches. So make sure that you don't have any caches that are tuned to have very large maximum sizes, or you may find that after a few hours you have used up all the available system memory, resulting in `OutOfMemoryExceptionS`.

### Setting the Interval parameter

During the test period you will have gained some experience with the relationship between the migration tool's **Interval** parameter, the size of the output files generated by the migration tool, and the amount of time it takes to generate them. You can then set the Interval parameter to whatever value you consider most suitable for your purposes. You can also vary the size of the **Interval** parameter during the migration process, so that old content is exported in large batches, but that once you get up to the present it is exported in frequent small batches.

### Running export/import in parallel

Export and import can run in parallel: you can simply periodically copy or move files from the export folder to the import folder.

### Running the systems in parallel

The export/import tools make it feasible to run the old and new systems in parallel during the migration process. Once the new copy of a publication is "complete" - the export/import process has reached the present, it is possible to continue editorial work on the old system, for example, but switch to the new system for publication purposes. Changes made on the old system will be exported once every minute and can be automatically copied to the new system for import.

#### **Automatic copying**

If you are intending running the old and new systems in parallel, then you will need to set up scripts for automatically copying files between the export and import folders.

#### **Cross-publishing issues**

If your site relies on cross-publishing, then you need to be able to establish relationships between content items in different publications. This means that the single-publication import sequence described in [section 2.2](#) must be modified slightly. Instead of importing all the content items in a single publication and then repeating the process, you need to import all the content items **in all inter-related publications** and then repeat the process.



---

## 3 Import

For general information about importing publications to version 5 of the Content Engine, see the **Escenic Content Engine Syndication Reference**. The information given here is specific advice on the most efficient way to import a migrated publication.

---

The **Escenic Content Engine Syndication Reference** describes the new syndication file format introduced in version 5, not the import/export format used in version 4. The Content Engine's import subsystem, however, supports both formats, so it is able to import data exported from version 4. The **Format Reference** section of this manual is therefore not of interest during migration, only those parts of the manual that deal with the import process. You should in any case not need to modify the content of the exported files in any way - simply re-import them according to the instructions given in this chapter.

---

There are some architectural differences between Content Engine versions 4 and 5, and the migration import process is designed to take account of these changes. During import, the following special actions are taken:

- Images, multimedia objects and links, which were represented by specialized objects in version 4, are converted to standard version 5 content items.
- References to images, multimedia objects and links in version 4 articles are correspondingly converted to standard version 5 relations.
- The catalogs used to organize the storage of images in version 4 are no longer supported in version 5. A special section tree is therefore created during import, which has the same structure as the source publication's catalog tree. This tree's root section is called `catalog root`, and it has a unique name of `migration_catalog`. Imported image and multimedia content items are assigned to the appropriate sections in this tree.

These conversions are carried out by migration import filters supplied with version 5 of the Content Engine. For a description of how to set up these filters see [section 3.1](#).

---

### 3.1 Migration Import Filters

Version 5.2.7.2 of the Content Engine includes a set of import filters designed to convert syndication files exported by the version 4 Migrator plug-in. The system is pre-configured to automatically create import folders for all defined publications on start-up. In other words, to create an import folder for a publication, you can:

1. Create an empty publication using the `escenic-admin` web application (see **Escenic Content Engine Server Administration Guide, section 2.1.10** for detailed instructions on how to do this).

2. Restart the Content Engine.

If you don't want to restart the Content Engine, then you can achieve the same results as follows:

1. Select the `escenic-admin` application's **Component Browser** option (see **Escenic Content Engine Server Administration Guide, section 2.1.13**).
2. Use the Component Browser to navigate to the `/com/escenic/syndication/xml/service/MigrationSetupService` component.
3. Click on **doStopService** to stop the service.
4. Click on **doStartService** to restart it.

By default the import folders are created in `/var/spool/escenic/migration`. The folder tree for a publication called `mypub` will look like this:

```
var
  spool
    escenic
      migration
        mypub
          archive
          error
```

With this set-up, you would copy the files to be imported into `/var/spool/escenic/migration/mypub`. Files that were successfully imported would be copied to `/var/spool/escenic/migration/mypub/archive` and files that failed to import successfully would be copied to `/var/spool/escenic/migration/mypub/error`.

If you want to move the import folders to a different location, then you can do so by changing the `MigrationSetupService`'s `migrationDirectory` property. To do this you need to edit `configuration-root/com/escenic/syndication/xml/service/MigrationSetupService.properties` in one of your configuration layers. For information about version 5 configuration layers and how to edit them, see **Escenic Content Engine Server Administration Guide, chapter 4**. Remember that if you change this setting then you will need to restart the Content Engine or the `MigrationSetupService` in order for the change to take effect.

The `MigrationSetupService` creates a `Reporter` for each publication import task. This is a component that monitors the import and reports on its progress. You can, if you wish configure these reporters to send mails when particular events occur. For further information, see **Escenic Content Engine Syndication Reference, section 2.3**.

---

## 3.2 Splitting Large Import Files

If a failure occurs while importing a syndication file, the whole file must be re-imported in order to ensure that no data is lost. Exported syndication files may in some cases be very large (large enough that importing them may take



many hours). It is therefore a good idea to split such large syndication files into smaller units before importing them. A failure during import will then be much less costly: only the failed import(s) will need to be repeated.

A tool for splitting large syndication files is included with the Content Engine. It is called `XMLSplit`. To split a file, enter:

```
$ java -cp classpath com.escenic.syndication.xml.util.XMLSplit filename number-of-elements
```

where:

- `classpath` is `engine-root/lib/xom-1.1.jar:engine-root/lib/engine-syndication-5.1-1.jar`.
- `filename` is the name of the syndication file you want to split.
- `number-of-elements` is the number of second-level elements you want each output file to contain.

For example:

```
$ java -cp engine-root/lib/xom-1.1.jar:engine-root/lib/engine-syndication-5.1-1.jar \
> com.escenic.syndication.xml.util.XMLSplit import.xml 100
```

If `import.xml` contains 950 second-level elements, then `XMLSplit` will output 10 files: 9 containing 100 second-level elements, and the 10th containing 50.

.....  
`XMLSplit` is in fact a general-purpose tool that can be used to split any large XML file, not just Escenic syndication files.  
 .....

---

### 3.3 HTML Markup

Version 4 of the Content Engine is very lenient with regard to HTML mark-up, and will accept many different "flavors" of HTML in HTML fields. This made it very easy to import content into the Content Engine, but could cause problems with display of the imported data in Content Studio and browsers, and also made it less easy to process and re-purpose content. Version 5 takes a different approach, and requires all the content in an HTML field to be valid XHTML mark-up.

This requirement obviously creates something of an obstacle to upgrading from version 4 to version 5, but is in the long-term best interests of most users of the Content Engine since it will:

- Increase the overall reliability of the publishing process.
- Make it easier to automatically process publication content, thereby increasing its potential value.

The importer automates the process of conversion to XHTML as far as possible, and for publications with "good quality" HTML content, the conversion may be completely invisible. All HTML field content is automatically passed through an HTML "clean-up" process during import, which silently converts the input stream to valid XHTML. This process, however, is not able to convert all possible variations of HTML to XHTML. When it encounters HTML

that cannot be converted it will fail. An error will be written to the log and the offending content item will not be imported. You will then need to correct the offending HTML and re-import the content item.

### 3.3.1 Configuring the HTML Clean-up Process

You can, if necessary configure the HTML clean-up process to deal with specific issues. To do this you need to edit `.properties` files in one of your configuration layers. For information about version 5 configuration layers and how to edit them, see **Escenic Content Engine Server Administration Guide, chapter 4**.

The process is controlled by the `HTMLCleanupSAXFilter` component, so you can configure it by editing `configuration-root/com/escenic/migration/filter/HTMLCleanupSAXFilter.properties` in one of your configuration layers.

The filter operates on the content of any `field` element in an input syndication file that has the attribute `type="html"`. The content of this kind of `field` element may **not** be XML markup: either all special characters must be escaped or all the content must wrapped in a `CDATA`. All HTML content exported by the version 4 Migrator is exported in this way.

The filter uses `FieldCleaners` to carry out the actual conversion. A field cleaner takes a string value as input and returns another string value containing the cleaned data. The `FieldCleaners` are chained together: output from one `FieldCleaner` is passed as input to the next `FieldCleaner`. The default `HTMLCleanupSAXFilter` has 3 field cleaners:

#### **PreTidyRegExFieldCleaner**

This `FieldCleaner` uses regular expressions to replace illegal/custom elements with valid XHTML equivalents.

You can configure this `FieldCleaner` by editing `configuration-root/com/escenic/migration/filter/PreTidyRegExFieldCleaner.properties` in one of your configuration layers. If you have custom tags that it is not be possible to convert in this way, a possible alternative is to create a custom field cleaner (see [section 3.3.1.1](#)).

#### **TidyFieldCleaner**

This field cleaner uses HTML Tidy (see <http://jtidy.sourceforge.net/>) to convert the input data to valid XHTML. Any markup that is not legal XHTML and cannot be converted will be discarded by this filter. This field cleaner plays a very important role in the whole process, since its output is guaranteed to be valid XHTML. This means that subsequent `FieldCleaners` can use XML manipulation techniques.

You can configure this `FieldCleaner` by editing `configuration-root/com/escenic/migration/filter/TidyFieldCleaner.properties` in one of your configuration layers.



#### **PostTidyRegExFieldCleaner**

This **FieldCleaner** uses regular expressions like **PreTidyRegExFieldCleaner**. Its main purpose is to revert some of the changes made by **PreTidyRegExFieldCleaner**. In particular, the special Escenic **reference** tag, used to represent in-line relations in syndication files, is converted to `<span x-escenic-type="reference">` by **PreTidyRegExFieldCleaner** in order to pass through the **TidyFieldCleaner**. **PostTidyReExFieldCleaner** converts it back to a **reference** tag.

You can configure this **FieldCleaner** by editing `configuration-root/com/escenic/migration/filter/PostTidyRegExFieldCleaner.properties` in one of your configuration layers.

### **3.3.1.1 Making Custom Field Cleaners**

If you cannot achieve the results you need using the default **FieldCleaners**, it is possible to write your own.

The **FieldCleaner** interface is very simple:

```
public interface FieldCleaner {
    String clean(String pFieldValue);
}
```

Assume, for example, that you need to run a XSL transformation on the field value to get your desired results. This can be achieved as follows:

1. Create an implementation of **FieldCleaner** called `com.example.XSLFieldCleaner` that runs the XSL transformation on the field value.
2. Compile and create a JAR file for this class.
3. Put the JAR file in your application server's classpath so that it can be referenced from the filter chain.
4. Create a properties file called `configuration-root/com/escenic/migration/filter/XSLFieldCleaner.properties` in one of your configuration layers with the following contents:

```
$class = com.example.XSLFieldCleaner # Plus any properties that you may need to set
```

5. Add your new **FieldCleaner** to the **HTMLCleanupSAXFilter** by appending the following line to `configuration-root/com/escenic/migration/filter/HTMLCleanupSAXFilter.properties` in one of your configuration layers:

```
fieldCleaner.400 = /com/escenic/migration/filter/XSLFieldCleaner
```

Note that the number 400 determines the **XSLFieldCleaner**'s position in the filter chain. Since running an XSL transformation requires valid XML input document, **XSLFieldCleaner** must run after **TidyFieldFieldCleaner**.

6. 5.Restart the Content Engine.

### 3.4 Avoiding Duplicate URIs

The Content Engine's default method for generating content item/article URIs is to append the content item's database ID to a prefix (`article` by default). This mechanism has not changed between version 4.x and version 5.2.7.2. When content items are migrated from version 4.x to version 5.2.7.2, they are assigned new database IDs. If no special action was taken, this would mean that all your publications' URIs would change as a result of the migration. In order to prevent this happening, the Migrator tool includes the old URI with every exported content item. This URI (`article123.ece`, for example) is imported into the system and stored with the content item as its URI. The new system will then always publish the content item as `article123.ece`, even though its database ID is (for example) `456`.

This mechanism ensures that all migrated content retains its original URI. However, the problem is not completely solved unless we also ensure that no newly-created content item in the new system ends up with the same URI. There are three possible ways you can avoid this problem:

- Set the starting ID number in your new database to a much higher number than the current ID number in your old system. If you are going to run the systems in parallel for any length of time, then you need to make sure that there is a large enough gap to allow for plenty of new content items being created on the old system.
- Change the prefix/suffix used for generating URIs to something different on the new system. You can do so by setting one or both of the following properties in your common configuration layer:

```
articlePagePrefix (article by default)
articlePageSuffix (.ece by default)
```

You must set the properties in *configuration-layer-root/ServerConfig.properties*. For general information on version 5 configuration layers, see **Escenic Content Engine Server Administration Guide, chapter 4**.

- Make a transaction filter that generates URIs for all your content. Making a transaction filter requires Java programming skills. For further information about making and using transaction filters, see **Escenic Content Engine Advanced Developer Guide, chapter 10**.

---

## 4 Upgrading Publication Web Applications

This chapter describes the changes that need to be made to 4.x applications in order for them to work with the 5.2.7.2 Content Engine.

---

### 4.1 Convert Publication Resources

To convert a version 4.x publication to version 5.2.7.2, you need to convert the publication resources to the new 5.2.7.2 resource file formats. This involves the following tasks:

#### Convert `article-type` resource to `content-type` resource

The new `content-type` resource format is completely different from the old `article-type` resource format. The version 5.2.7.2 installation includes a tool (an XSL transformation) that automates this conversion. You will find the tool in your installation's `/opt/escenic/engine/contrib/upgrade` folder. It consists of the following files:

`article-to-content-types.xsl`: the actual transformation

`xslt`: a script to run the transformation on Unix/Linux

`xslt.bat`: a script to run the transformation on Windows

To run the tool on either platform, add the folder containing the tool to your path, then enter:

```
xslt article-to-content-types.xsl article-type-path >content-type
```

where `article-type-path` is the path to your input `article-type` resource.

Note that this transformation carries out a one-to-one translation from 4.3 concepts to 5 concepts. The resulting `content-type` resource may not be completely valid, and may therefore be rejected with validation errors the first time you try to upload it. If this occurs, correct the reported validation errors and try again.

---

There are some steps you must perform manually while doing this conversion. Those are given below.

- If you have some `<icon>` element inside your `article-type` resource definition then those icon files should be accessible to the system via http. And the url of this resource should be added manually in the new `content-type` resource. Suppose you had one icon file named `bigIcon.gif` in the older system as publication resource and your `article-type` definition has `<icon>bigIcon</icon>`. In order to make it work you need to upload it in some web location and put the url inside icon definition like this `<ui:icon>http://<mysite>/statics/bigIcon.gif</ui:icon>`.
- If you have forum plug-in installed in the version 4.x system then the converted `content-type` resource will contain one Posting

`content-type`. You need to manually add the following field definitions to that `content-type` if not present.

```
<field mime-type="text/plain" type="basic" name="type">
  <ui:label>Type</ui:label>
  <ui:hidden/>
</field>
<field mime-type="text/plain" type="basic" name="email">
  <ui:label>Email</ui:label>
  <ui:hidden/>
</field>
<field mime-type="text/plain" type="basic" name="state">
  <ui:label>State</ui:label>
  <ui:hidden/>
</field>
<field type="boolean" name="processed">
  <ui:label>Processed</ui:label>
  <ui:value-if-unset>true</ui:value-if-unset>
  <ui:hidden/>
</field>
<field type="boolean" name="accepted">
  <ui:label>Accepted</ui:label>
  <ui:value-if-unset>false</ui:value-if-unset>
  <ui:hidden/>
</field>
```

---

### Convert `layout-resource` to `layout-group-resource`

Version 5.2.7.2 offers a great deal of new functionality with regard to section page structure, so you may decide to combine a page redesign with the upgrade, in which case you can just make a new `layout-group` resource from scratch. Note, however, that whichever method you use, you will need to upload your original 4.x `layout` resource along with the new `layout-group` resource when creating the version 5.2.7.2 publication. Also, you need to copy "layouts.dtd" from 4.x to the location of 5.x server, where the layout.xml file is located. If in the 4.x layout resource you have some group or area names containing some illegal characters than uploading the new `layout-group` resource might fail. You need to manually correct those names after the conversion.

### Upgrade `image-versions` resource (optional)

Version 5.2.7.2 provides new image cropping and sizing functionality that is more flexible than the old image versions functionality, therefore you don't **need** an `image-versions` resource. If, however, you want to continue using the old functionality, then you need to add a namespace declaration to the root element of the `image-versions` resource:

```
<imageDef xmlns="http://xmlns.escenic.com/2008/image-versions">
  ...
</imageDef>
```

---

Element names in the version 5.2.7.2 XML resource files must always be lower-case. Version 4.x, however, was more lenient and in many cases case-insensitive. The `article-type` resource, for example, could contain both `panel` and `Panel` elements. You must ensure that all element names are lower case before uploading your converted resources, or they will be rejected. The `article-to-content-types.xsl` conversion tool will not do this for you.

---

## 4.2 Handle Relation Types

Relation type handling has changed in version 5. In version 4.x it was possible to have implicit relation types for any field in an article. In version 5 this is no longer the case: only relation types that are explicitly defined in the `content-type` publication resource are allowed. In addition, relation types names are resolved in the same namespace as fields, so a relation type may not have the same name as a field.

In order to avoid naming problems, when the Migrator encounters such ad-hoc relations in exported data, it auto-generates a relation type name by adding the suffix `_RELATED` to the field name.

These auto-generated relation types, however, are not included in the converted `content-type` resource generated by `article-to-content-types.xml`.

This means that if your 4.x publication made use of such undefined relation types, then you will need to add definitions for them to the `content-type` resource created by `article-to-content-types.xml`. If, for example, your old articles contained relations to fields called `title`, then you would need to add a relation called `title_RELATED` to your `content-type` resource, as shown below:

```
<field type="basic" mime-type="text/plain" name="title">
...
</field>
<relation-type-group name="news-relations">
  <relation-type name="title_RELATED"/>
</relation-type-group>
```

Fields and relation types were allowed to have identical names in 4.x article type definitions. This is not normally the case, however, for version 5.2.7.2 `content-type` definitions. If your 4.x article type definitions contain identical field and relation type names, then errors will occur when you try to upload the converted `content-type` resource. You can prevent this happening by setting the `/neo/io/services/validators/ContentTypeValidator` component's `enableSameNameForFieldAndRelationType` property to `true`. By default this property is set to `false`.

## 4.3 Upgrade WEB.XML

You must remove the following elements from your publications' `WEB.XML` files:

```
<filter>
  <filter-name>imageGenerator</filter-name>
  <filter-class>com.escenic.presentation.servlet.multimedia.ImageVersionGenerator</filter-class>
  <init-param>
    <param-name>oncePerRequest</param-name>
    <param-value>true</param-value>
  </init-param>
  <init-param>
    <param-name>imageVersionDefinitions</param-name>
    <param-value>/escenic/image-version</param-value>
  </init-param>
</filter>
<filter-mapping>
  <filter-name>imageGenerator</filter-name>
```

```
<url-pattern>/multimedia/*</url-pattern>
</filter-mapping>
```

You should also add the following elements:

```
<filter>
  <filter-name>TopFilter</filter-name>
  <filter-class>com.escenic.servlet.TopFilter</filter-class>
  <init-param>
    <param-name>shared-instance</param-name>
    <param-value>nursery://global/neo/io/reports/Top</param-value>
  </init-param>
</filter>

<filter-mapping>
  <filter-name>TopFilter</filter-name>
  <url-pattern>/*</url-pattern>
  <dispatcher>REQUEST</dispatcher>
  <dispatcher>FORWARD</dispatcher>
  <dispatcher>INCLUDE</dispatcher>
</filter-mapping>
```

The above elements are required to provide support for the **escenic-admin** web application's **Top** page (see **Escenic Content Engine Server Administration Guide, section 2.1.4**). If you do not add the above elements, then this new statistics utility will not work.

## 4.4 Upgrade Templates

- The `<relation:media>` and `<relation:link>` tags have been withdrawn, so any of your publication templates that make use of them must be rewritten. For general advice on how to render relations in version 5 templates, see **Escenic Content Engine Template Developer Guide, section 6.3**.
- `<template:insert type="element">` has been withdrawn, so any of your publication templates that make use of this must be rewritten. You will need to traverse the object hierarchy to get to the required group
- `<article:field transform="xx">` is deprecated and should be replaced with `<article:renderField>`, which deals with in-line images in a better way.

## 4.5 Upgrade Java Code

- You should recompile any custom Java code you have written.
- The following classes have been removed from the Content Engine's public API, so any custom Java code you have written which makes use of them will need to be modified:

```
neo.xreditsys.api.Image
neo.xreditsys.api.ImageTransaction
neo.xreditsys.api.Link
neo.xreditsys.api.LinkTransaction
neo.xreditsys.api.Media
neo.xreditsys.api.MediaTransaction
neo.xreditsys.api.MultiMediaVersionType
neo.xreditsys.api.MultimediaVersion
```



`neo.xreditsys.api.media.AbstractImageVersionGenerator`

Classes have also been removed from the following internal packages:

`neo.xreditsys.api.services`  
`neo.xml`  
`neo.xreditsys`  
`neo.xreditsys.api.media`  
`neo.xreditsys.content.article`  
`neo.xreditsys.content.link`  
`neo.xreditsys.content.multimedia`  
`neo.xreditsys.content.xmlentity`  
`neo.xreditsys.persistent`  
`neo.xreditsys.presentation`