

# Link Management

by Christian Egli

## 1. Introduction

Link Managements deals with internal links, i.e. documents that refer to other documents within the same publication. These links might have to be changed.

1. if a document not yet live,
2. if it is withdrawn from live or
3. if its document-id has changed because it is moved to a different location within the site tree.
4. references in authoring have to link to documents in authoring, however once they are published they need to refer to documents in live

These four cases have to be dealt with seperately.

### 1.1. A document is not yet live

This case can happen if the user tries to publish a document which has a reference to another document which has not been published yet. The reference will be stale as the refered document is not in the live area yet. A warning will be issued during the publishing process.

### 1.2. A document is withdrawn from live

If a document which is has references to it is withdrawn from the live area the references will be stale, as the refered document is no longer available in the live area. A warning will be issued during the deactivation process.

### 1.3. A document-id changes

If a document is moved within the site tree such that it changes its document-id (e.g. cut a document and paste it somewhere else in the hierarchy in the site area) then all references to this document have to be changed. This is done transparently in the course of the paste.

### 1.4. Rewrite internal links in live

Internal links refer to documents in authoring as long as they are not published. However as soon as they are published, i.e. reside in the live area their references have to go to documents in the live area. A transformer takes care of rewriting the internal links.

## 2. Implementation

The implementation is pretty straight-forward and is mostly handled in XSPs and associated helper classes. The heavy lifting us done is the class `Grep` and the helper class `DocumentReferencesHelper`. The `Grep` class can traverse the repository and find references to the current document or can also find references from the current document to other documents. A transformer (`SimpleLinkRewritingTransformer`) is used to rewrite the internal links in the live area.

### 2.1. A document is not yet live

This is implemented as an extension to the `publish xsp`. It queries the `DocumentReferencesHelper` (using the `getInternalReferences` method) to ask if there are references from the current document to other documents which have not been published yet. The `DocumentReferencesHelper` in turn uses the `Grep#findPattern` method to search the current document for patterns of a reference. The pattern is defined in `DocumentReferencesHelper#getInternalLinkPattern`.

### 2.2. A document is withdrawn from live

This is implemented as an extension to the `deactivate xsp`. It queries the `DocumentReferencesHelper` (using the `getReferences` method) to ask if any other documents refer to the current document. The `DocumentReferencesHelper` in turn uses the `Grep#find` method to search the repository for documents which contain a patterns of a reference. The pattern is defined in `DocumentReferencesHelper#getReferencesSearchString()`.

### 2.3. A document-id changes

This is implemented with an ant task (`org.apache.lenya.cms.ant.LinkRewriteTask`) which traverses the repository and pipes all documents through an XSLT stylesheet (`unizh/pubs/unizh/xslt/linkRewrite.xsl`) to modify all references to the document that changed its document-id.

## 2.4. Rewrite internal links in live

The

`org.apache.lenya.cms.cocoon.transformation.SimpleLinkRewritingTransformer` transformer takes care of rewriting internal links to ensure they refer to the appropriate area.

## 3. Involved classes, XSPs and XSLTs

The following classes, XSPs and XSLTs are involved in link management:

**`org.apache.lenya.cms.publication.xsp.DocumentReferencesHelper`**

A helper class for the publish and deactivate xsp. Defines the regular expressions for internal links. Has methods to determine all references from the current document to other documents (`getInternalReferences`) and to determine all references from other documents to the current document (`getReferences`).

**`org.apache.lenya.search.Grep`**

User by `DocumentReferencesHelper` to search for patterns in a file or in a directory tree.

**`org.apache.lenya.cms.ant.LinkRewriteTask`**

An ant task that upon change of a document-id pipes all documents of the repository through a XSLT stylesheet which rewrites internal links that were referring to the old document-id to refer to the new one.

**`org.apache.lenya.cms.cocoon.transformation.SimpleLinkRewritingTransformer`**

A transformer that rewrites internal links for the appropriate area.

**`$publication-id/config/tasks/targets.xml`**

Defines the `move-and-rewrite` target which handles the link rewriting in the case of a paste, i.e. when a document-id has changed.

**`unizh/pubs/unizh/xslt/linkRewrite.xsl`**

The XSLT transformation used by `LinkRewriteTask` to actually rewrite the internal links.

**`src/webapp/lenya/content/publishing/screen.xsp,`**

**`src/webapp/lenya/xslt/publishing/publish-screen.xsl`**

Query the `DocumentReferencesHelper` to display a warning in case the current document contains references to documents which have not been published yet.

**`src/webapp/lenya/content/info/deactivate.xsp,`**

**`src/webapp/lenya/xslt/info/deactivate.xsl`**

Query the `DocumentReferencesHelper` to display a warning in case there are links to the current document which is about to be deactivated.