

# OverviewLenyaPublication

- Target-audience: **intermediate**
- Target-role: **Architect** [more](#)
- Target-release: 1.2.2
- Doc-status: **draft**

What you will get from this page

A presentation of the default Lenya publication internals. This should help you to build a [PublicationFromScratchHowTo](#).

---

## The big picture

Here an overview of the default publication (to be found in \$LENYA-HOME/pubs). The most important files and directories are set in bold. The links will bring you to the detailed description.

1. **config**; configuration files
  - a. **ac**; access control (identity, group and roles; policies)
  - b. **doctypes**; doctypes.xconf, samples and schemas
  - c. **menus**; generic.xsp and links.xsp
  - d. publishing; publisher.xconf
  - e. search; lucene configuration
  - f. **tasks**; tasks definitions in tasks.xconf and their properties in targets.xml
  - g. workflow; workflow.xml
    - **publication.xconf** XPatch files for Cocoon.xconf (to declare input modules, etc)
2. **content** ---> **your content here (XML Files)**
  - a. **archive**; archived files
  - b. **authoring**; authoring files
  - c. **live**; live files (published)
  - d. rcml; revision control, for managing checkin/checkout of files
  - e. rcbak; rc-backup of files. For archiving of back up files see [ClearDown](#)
  - f. trash; where deleted files are stored
  - g. workflow; holds a log history of the workflow
3. **java**; publication specific java classes or libs
4. **lenya**; files replacing core files
  - a. **content**
    - i. authoring
    - ii. publishing
  - b. **resources**
  - c. **xslt**
5. **logs** (build) logs
6. **resources**; static resources, like css and images
  - a. shared
    - i. **css**
    - ii. **images**
7. **test**; unittests (antteater and webtest)
8. **tools**; tools, currently there's a clone app
9. **work**; cache and working directory
10. **xslt**; XSLT stylesheets

- **doctypes.xmap**
  - **menus.xmap**
  - **parameter-doctype.xmap**; ??
  - **publication.xml**; meta-information about the pub. is shown when you enter the publication
  - **publication-sitemap.xmap**; main sitemap
  - **publication-tests.xml** publication tests
  - **sitemap.xmap**; only mounts publication-sitemap.xmap
- 

## The publication dissected

### sitemap.xmap and publication-sitemap.xmap

Let's start with the file `sitemap.xmap`, which only mounts `publication-sitemap.xmap`. In `publication-sitemap.xmap`, the

There's a good explanation of this pipeline [here](#) under the header "The Lenya pipeline"

## config

Where the configuration files of the publication are stored.

### config/ac

stands for [AccessControl](#) also see the [basic terms](#)

#### config/ac/password

where the identity, groups and roles are stored

#### config/ac/users

Have a look at our dear Alice (`alice.iml`) = identity markup language).

```
<identity id="alice" class="org.apache.lenya.ac.file.FileUser">
  <name>Alice</name>
  <description></description>
  <email>alice@lenya.org</email>
  <password type="md5">8e07dafd13495561db9063ebe4db4b27</password>
  <groups>
    <group>editor</group>
    <group>reviewer</group>
  </groups>
</identity>
```

Then create a new user (eg. ruppert) in the admin-area of your lenya-publication. It should then appear as `ruppert.iml`

#### config/ac/groups

groups saved as gml (group markup language)

```
<group class="org.apache.lenya.ac.file.FileGroup" id="reviewer"/>
```

to hardcode a new group, create a new `.gml` file (eg. `designer`) and replace the `id` with "designer". You could then add this group to Alice by editing `alice.iml`.

```
<groups>
  <group>editor</group>
  <group>reviewer</group>
  <group>designer</group>
</groups>
```

#### config/ac/roles

finally, the roles are saved as `.rml` (role markup language), eg. `review.rml`

```
<role class="org.apache.lenya.ac.file.FileRole" id="review"/>
```

TODO: can we hardcore new roles the same way as groups ?

### config/policies

1. admin
  - subtree-policy.acml
  - a. users
    - i. alice.html
      - subtree-policy.acml
    - ii. anne.html
      - subtree-policy.acml
    - iii. lenya.html

- subtree-policy.acml
- iv. ludeditor.html
  - subtree-policy.acml
- v. renaud.html
  - subtree-policy.acml
- vi. walter.html
  - subtree-policy.acml
- 2. archive
  - subtree-policy.acml
- 3. authoring
  - subtree-policy.acml
- 4. index.html
  - url-policy.acml
- 5. info
  - subtree-policy.acml
- 6. introduction.html
  - subtree-policy.acml
- 7. live
  - subtree-policy.acml
- 8. trash
  - subtree-policy.acml
- url-policy.acml
- usecase-policies.xml

## config/doctypes

- doctypes.xconf
- samples
- schema all dublincore and xhtml schemas (.rng)

## config/menus

To add a new menu item like "New [MyDocType](#)" to the File menu, you add a line like to yourpub/config/menus/generic.xsp:

```
<xsp:logic>
{
  if (Publication.ARCHIVE_AREA.equals(area) ||
      Publication.TRASH_AREA.equals(area)) {
    <item><i18n:text>New Document</i18n:text></item>
  }
  else {
    <item uc:usecase="create" uc:step="showscreen"
          href="?doctype=xhtml"><i18n:text>New Document</i18n:text></item>
    <item uc:usecase="create" uc:step="showscreen"
          href="?doctype=yourDocType"><i18n:text>New yourDocType
          </i18n:text></item>
    <!-- add this line, changing the parameters for your doctype -->

  }
}
</xsp:logic>
```

## config/publishing

## config/search

## config/tasks

- targets.xml
- tasks.xconf
- see [the task concept](#)

## config/workflow

- workflow.xml
- see [the workflow configuration](#)

## config/publication.xconf

contains publication configuration, eg. [language selection](#) and [proxying](#)

---

## content

### content/archive

- sitetree.xml

### content/authoring

#### 1. concepts

- index\_de.xml
  - index\_en.xml
    - 1. doctypes
  - index\_de.xml
    - 1. xhtml-document
      - index\_de.xml
      - index\_en.xml
    - 2. features
      - index\_de.xml
      - index\_en.xml
        - 1. index
      - index\_de.xml
      - index\_en.xml
        - 1. intro
      - index\_en.xml
        - 1. ludicotest
      - index\_en.xml
        - sitetree.xml
        - 1. tutorial
      - index\_de.xml
      - index\_en.xml
    - 1. new\_doctype
      - index\_en.xml

### content/live

- 2. index
  - index\_en.xml
    - 1. ludicotest
  - index\_en.xml
    - sitetree.xml

### content/rcbak

- 1. content
- 1. authoring
  - a. index
  - index\_en.xml.bak.1114085410863
    - 1. ludicotest
  - index\_en.xml.bak.1113924105953
    - index\_en.xml.bak.1113924375441
    - index\_en.xml.bak.1113924386797
    - index\_en.xml.bak.1113992891077

### content/rcml

- 1. content
- 1. authoring
  - a. index
  - index\_en.xml.rcml
    - 1. intro
  - index\_en.xml.rcml
    - 1. ludicotest
  - index\_en.xml.rcml

### content/trash

- sitetree.xml

## **content/workflow**

## **content/doctypes.xmap**

---

### **java**

That's where you put your publication specific classes.

By default, there are 2 classes in the package org.apache.lenya.defaultpub.cms.task (publish.java and deactivate.java)

TODO: who calls those classes?

---

### **lenya**

Files replacing core files.

#### **lenya/content**

1. authoring
  - create-language.xsp
  - create.xsp is displayed when you create a new content.
  - upload.xsp is displayed when you upload a new content.
2. publishing
  - publish-screen.xsp

#### **lenya/resources**

3. i18n
  - cmsui.xml
  - cmsui\_de.xml
  - cmsui\_es.xml
  - cmsui\_fr.xml
  - cmsui\_it.xml

#### **lenya/xslt**

4. authoring
    - create-language.xsl
    - create.xsl
  5. formeditor
    - homepage.xsl
    - xhtml-common.xsl
    - xhtml.xsl
  6. navigation
    - tabs.xsl
- 

### **logs**

Logs. Lenya stores here the logs of the publication-build process.

---

### **ressources**

---

### **test**

anteater and webtest files (tests.xml) come here

---

## tools

currently, there is a clone utility.

---

## work

cache-directory, so nothing very interesting here.

1. cache
  2. editor
  3. oneformeditor
  4. search
    - a. lucene
- 

## xslt

- homepage2xhtml.xsl
  - lenya-header.xsl
  - links2xhtml.xsl
  - page2xhtml-homepage.xsl
  - page2xhtml-links.xsl
  - page2xhtml-xhtml.xsl
  - page2xhtml.xsl
  - toDoc.xsl
  - upload-generic.xsl
  - upload.xsl
  - xhtml2xhtml.xsl
- 

## doctypes.xmap

TODO

## publication.xml

TODO

## menus.xmap

TODO

## parameter-doctype.xmap

TODO

## publication-tests.xml

TODO

## usecase-bxeng.xmap

TODO

originally contributed by [RenaudRichardet](#)