

[EN] 03. The Structure of the Reader Application of the dLibra System

The Structure of the Reader Application of the dLibra System

- [The Basic Concepts Related to the Functioning of the Reader Application](#)
- [A Description of the Configuration Files of the Reader Application of the 6.0 Version of the dLibra System](#)
 - [The Configuration Files in the dlibra-webapp/WEB-INF/conf Directory](#)
- [A Description of the Structures at dlibra-webapp\style](#)
- [Additional Elements](#)

The Basic Concepts Related to the Functioning of the Reader Application

Pages – the reader application consists of many pages (views), for example, the home page, the object page, or the contact page. They are the basis of navigation. Every page has its own, unique identifier, for example:

Identifier	Page	Sample address
main	home	http://demo.dl.psnc.pl/dlibra
publication	object information page	http://demo.dl.psnc.pl/dlibra/ publication /1000
contact	contact page	http://demo.dl.psnc.pl/dlibra/ contact

Each of those pages is defined in the **pages.xml** file by section **<page>...</page>**, for example:

Identifier	Sample definition
main	<div>pages.xml - main</div> <pre><page name="main" layout="home_layout" inherits="base-page"> ... </page></pre>
publication	<div>pages.xml - publication</div> <pre><page name="publication" layout="home_layout" inherits="base-page"> ... </page></pre>
contact	<div>pages.xml - contact</div> <pre><page name="contact" layout="home_layout" secure="true" inherits="base-page"> ... </page></pre>

Page tags have three different fields:

name – the identifier of a page, for example: “main”, “publication”, or “contact”;

layout – the name of a layout (a graphic layout of a page) in which all components of a page will be set; and

inherits – the identifier of a page (it can be abstract) from which the page which is being defined will take components (apart from its own components); for example: the “contact” page is defined in the “contact” section and contains the ContactComponent component, but the “inherits” field has the “base-page” value, so the contact page will also contain all the components defined for the “base-page” section;

Components – correspond to particular elements of a page; for example, the section with recently added objects on the home page is a separate component; components can be displayed in various places on a page; the components to be displayed on a page ought to be specified in the descriptor of the page, in the pages.xml file; the method of defining a component is presented below: The “component” tag is characterized with the use of the following attributes:

pages.xml - component

```
<component name="...">
    <place>...</place>
    <position>...</position>
</component>
```

The “component” tag is characterized with the use of the following attributes:

name – the full name of a component;

place – the name of the position of a component in a layout; and

position – the position in which the component should be put in a place; if there is another component in the same place, the position must have a unique value.

One example of a component is the component on the contact page. That page is defined in the page contact element. It uses the **home_layout** layout. It has a component which is defined in the following manner:

pages.xml

```
<components>
    <component name="pl.psync.dlibra.web.comp.pages.components.ContactComponent">
        <place>main</place>
        <position>1</position>
    </component>
</components>
```

It means that the content of the **ContactComponent** component in the **/WEB-INF/components/templates/ContactComponent.vm** file is rendered in the **/WEB-INF/layouts/templates/home_layout.vm** distribution template, in a loop:

home_layout.vm

```
#foreach( $comp in $main)
    ${comp.RenderedTemplate}
#end
```

Every component consists of three elements:

- a Java class (for example, **pl.psync.dlibra.web.comp.pages.components.IndexSearchComponent**),
- a template file, which is named the same as the Java class, for example, **IndexSearchComponent.vm** ; and
- files with various language versions, for example, **IndexSearchComponent_xx.xml**).

in version 6, all components are located in the **dccore-webapp-components-6.x.x.jar** archive, in the **/WEB-INF/lib** directory. The Java class is in the directory which corresponds to its package name (in this case, **/pl/psync/dlibra/web/comp/pages/components/**), and the remaining files of the component – in the root directory of that archive.

The default files from the JAR archive (**dccore-webapp-components-6.x.x.jar**) can be “covered” with the user’s own, changed files. For that purpose, it is best to unpack the template files which are to be changed, to an appropriate directory. In the case of **.vm** files, it is the **/WEB-INF/components/templates** directory, and in the case of **.xml** language files, it is the **/WEB-INF/components/resources** directory.

Files unpacked in that way can be changed, and the original content of the files left in the **JAR** archive will be overwritten with the new data. The changes in the ***.vm** files will be visible immediately after they have been saved. The changes in the ***.xml** files will also be taken into account immediately if they were already unpacked when the Tomcat server was being started up. Otherwise, the Tomcat server should be restarted.

Moreover, in some component templates, macros defined in the **components_library.vm** or **layout_library.vm** file are used. File **componentslibrary.vm** is in the **dccore-web-components-6.x.x.jar** archive. The user’s own macros and modified macros should be placed, appropriately, in files **/WEB-INF/components/templates/custom_library.vm** or **/WEB-INF/layout/custom_templates/custom_layout_library.vm**). The changes in those files will also be taken into account immediately after they have been saved if the macro edited by the user was available when the Tomcat server was being started up.

A Description of the Configuration Files of the Reader Application of the 6.0 Version of the dLibra System



It should be remembered that configuration files should be edited with the use of an editor which supports UTF-8 encoding and does not place the BOM tag at the beginning of a file – for example, with the use of the free JEdit editor.

Below, there is a description of the most important configuration files for the reader application of the dLibra system. Changes made in some files require a restart of the reader application. Those files are marked with **(R)** directly after their names. Files not included in the breakdown below should not be modified without prior consultation with the creators of the dLibra system.

- **actions.xml** – that file contains additional configuration parameters for actions, including:
 - the addressee email address for messages sent by users through the contact form and the format of those messages (plain text or HTML);
 - the configuration of the protections of the mechanism which allows users to propose key words (only for implementations with dedicated solutions for that):
 - whether the proposals are to be rejected or accepted by default,
 - the maximum and minimum length of the proposals of key words, and
 - the maximum number of proposals which can be entered by a user in a given period of time (by default, the maximum is five proposals during 30 minutes);
 - the maximum and minimum length of the login entered by a user during registration; and
 - the metadata fields which the user will be able to use for sorting search results.
- **components.xml** – additional configuration parameters for particular components; in that file, the following elements should be changed:
 - the collection identifier and “Recommended”,
 - for the index component – the list of tokens which, as prefixes of value indexes, will not be taken into account during sorting;
 - the maximum length of the search query;
 - the possibility of displaying a form with the question about access to a protected publication;
 - the possibility of filtering by collections, group publications, and accessibility in the search results;
 - the names of the attributes which should not be searchable;
 - the number of elements displayed on the home page for the recommended, recently added, and most popular publications and news;
 - identifiers of collections which are not to be displayed in the collection structure and whether the collection structure is to be a complete tree;
 - the names of the attributes which are to be displayed in the index component;
 - the email address of the person responsible for the administration of the OAI repository; and
 - the basic attributes in the digital object information view; the possibility of identifier an OAI identifier;
- **ignored_ips.txt** i **ignored_agents.txt** (R) – those files are related to computing the visit statistics; in the first file, the IP addresses to be counted (for example, the computers on which editors work) should be entered; every entry is treated as a regular expression, so complete ranges of IP addresses can be entered; the file contains a sample entry for 127.0.0.1; before adding a new regular expression, it is worth checking if it is correct, with the use of free tools available online, for example, <http://regexpal.com/>.

The **ignored_agents.txt** functions in a similar manner: regular expressions for the “user-agents” field should be placed here. The field is a part of the heading of a HTTP request. It is a sequence of characters which identify the program used for browsing a page (for example, an Internet browser or a browser bot). If the value of the sequence is adapted to the given model, the programs identifying with the use of that sequence will not be counted for the statistics of visits and impressions of publication content. The file contains a sample entry for Googlebot. The list of “user agents” identifiers can be found online.

- **pages.xml** – that file contains the definition of the pages displayed by the reader application of the dLibra system; it must be modified when:
 - the order of the components displayed on a given page is changed;
 - a new component is being added to a page; or
 - a new page is being created in the reader application; the issues related to modifying that file are discussed in detail in the “Adjusting the Look of the Reader Application” and the “Frequently Asked Questions” chapters;
- **pages_titles.xml** (R) – in that file, the titles and subtitles of particular pages are determined;
- **templates.properties** (R) – contains the configuration parameters available from the level of all templates;
- **periodic.xml** (R) – that file contains definitions of periodic tasks performed by the reader application; the following elements are configured in it:
 - the time of creating a backup of JCR, the path in which the backup is to be stored, and the number of stored previously created backups;
 - the mechanism of importing news from an external RSS;
 - the time of generating and saving, and of generating statistics presentation views; identifiers of the attributes on the basis of which attribute value statistics are created;
 - the time of sending newsletters with recently added publications;
 - the location of the directory in which information about reader application users’ search queries are saved; and
 - the time and mechanism of creating the site map.;
- **files dlibra-webapp\WEB-INF\components\resources\WEBAPP_xx.xml**xml – where xx means the language version (for example, .pl); that file contains the most frequently modified text labels for every installed language version;
- **dlibra-webapp\WEB-INF\components\templates\custom_library.vvm** – contains additional VTL language macros for components, created by the administrators of the given library; information about creating VTL language macros can be found on the web pages of the Velocity project;

- **dlibra-webapp\WEB-INF\layout\custom_templates\custom_layout_library.vvm** – contains additional VTL language macros for distribution templates, created by the administrators of the given library; information about creating VTL language macros can be found on the web pages of the Velocity project;
- files **dlibra-webapp\WEB-INF\layout\resources\layout_xx.xml** – where xx means the language version (for example, .pl); that file contains text labels used in distribution templates;
- **dlibra-webapp\WEB-INF\formats** directory – contains the configurations of all extensions related to presenting publication content; for more information, see section 03. [The Extensions of the Reader Application](#).

The Configuration Files in the dlibra-webapp/WEB-INF/conf Directory

The reader application uses its local database which is completely unrelated to the database used by the dLibra server. The configuration of that base consists of, apart from the two files discussed below, `jcr-init.properties` and `jcr-repository.xml` files. However, they should never be modified.

- **jcr-init-desc.xml** (R) – the specification of the basic content (default values) put in the database during the first start-up of the reader application.
- **jcr.properties** (R) – configuration of the basic access parameters for the database of the reader application. The working directory of the reader application contains a path to the database; that path should be updated when the directory is moved.
- **services.properties** (R) – the parameters for connecting to the dLibra server. The dLibra server contains the server address; that address should be updated when the server is moved to another machine.
- **smtp.properties** (R) – access parameters for the SMTP server. Those parameters are used by the reader application for sending emails.
- **cache.properties** (R) – contains the cache settings of the reader application. When cache is used, the functioning of the reader application is significantly accelerated. Cache must be used in the case of libraries containing more than 1000 publications.
- **exlibris.xml** – the content of that file contains the configuration of the bookplate mechanism. The bookplate configuration is discussed in detail in the “Configuring the Reader Application” section.
- **pubcreator.properties** – that file contains the basic parameters of the self-archiving mechanism. The self-archiving configuration is discussed in detail in the “Configuring the Reader Application” section. It applies to dedicated implementations with the “self-archiving” function switched on.
- **pubc-metadata.xml** – contains the configuration of the attribute form for the “self-archiving” functionality. That topic is discussed in detail in the “Configuring the Reader Application” section. It applies to dedicated implementations with the “self-archiving” function switched on.

A Description of the Structures at dlibra-webapp\style

Most graphics files and CSSs used in the reader application are in the **dlibra-webapp\style** directory.

Additional Elements

Apart from standard references to pages defined in the reader application, there are also resources available from the main level:

- <http://demo.dl.psync.pl/Content> – responsible for access to publication content;
 - calling `/Content/2952/` will cause a transition to the content of the main file of the edition with the 2952 identifier;
 - calling `/Content/2952/directory.djvu` will cause a download of the .djvu file from the edition with the 2952 number;
- <http://demo.dl.psync.pl/zipContent> – responsible for access to publication content in the form of ZIP archives; calling `/zipContent/2952/` will cause a download of the content of the whole edition with the 2952 number, as a ZIP file;
- <http://demo.dl.psync.pl/image> – responsible for access to edition thumbnails; calling `/image?id=2940` will cause a download of the thumbnail of the edition with the 2940 number;
- <http://demo.dl.psync.pl/ka> – responsible for encrypting protected publications;
- <http://demo.dl.psync.pl/dlibra/dlibra-app.jnlp> – entering that address to a browser will start up the Editor and Administrator Application;
- <http://demo.dl.psync.pl/dlibra/publication/81/content> – a reference to the content of the publication with the 81 identifier; if the publication has more than one edition, a list of the editions of that publication will be displayed; in the case of a group publication, the structure of the group publication will be displayed;
- <http://demo.dl.psync.pl/dlibra/publication/81/> – a reference to the description of the publication with the 81 identifier; if the publication has more than one edition, a list of the editions of that publication will be displayed;