User Guide for version 4.0 of the dLibra system

Poznań Supercomputing and Networking Center
Edited by Tomasz Parkoła

User Guide for version 4.0 of the *dLibra* system

by Tomasz Parkoła

Table of Contents

Chapter 1. Introduction

1.1. *dLibra* - digital library framework

The *dLibra* Digital Library Framework is a set of tools that allow:

- Storage of various digital objects (e.g. scientific articles, reports, manuals, HTML pages, PDF documents, audio/video recordings)
- Management of digital objects stored in a library (e.g. modifying objects, grouping objects, assigning objects to thematic categories, management of user rights assigned to objects)
- Access to digital objects (e.g. viewing objects on web pages, searching for objects, securing objects from copying, temporal access)
- Management of a digital library (e.g. creating groups of users, users' rights management)
- Cooperation with other information systems using publicly available standards (e.g. OAI-PMH, RSS)

dLibra system is composed of three basic elements:

- Server - key element of the system; responsible for every operation performed on digital library,
- WWW pages - reader's application; allow users to access digital library resources (e.g. books),
- Editor and administrator application - desktop program composed of two applications - editor's and administrator's which are used to manage library resources (e.g. adding new elements) and the library itself (e.g. management of digital library thematic collections).

These three connected elements make *dLibra* system - platform for building digital libraries.

A set of discussion forums are available on *dLibra* project web page. These forums concern both technical help for *dLibra* system (editor and administrator application, reader application, *dLibra* system installation and configuration, etc.) and general topics concerning digital libraries and *dLibra* system. Editor application forum and administrator application forum is reserved for editor and administrator application users. All the forums can be found under http://www.dlibra.psnc.pl/forum. If you have any questions/problems/suggestions we invite you to post it on the forum.

1.2. *dLibra* system users

Users of the *dLibra* system may be divided into three common types:

- Administrator - manages digital library (creates users and groups of users, grants rights to users and groups), can create and modify library structure, can use all of the *dLibra* system applications (administrator, editor, WWW) but generally uses administrator application,

- Editor - creates digital library (adds new digital objects to library, creates or imports description for objects, groups objects, generally has limited access to administrator application, uses editor application,

- Reader - *dLibra* system user which has access only to web pages (WWW) placed on the Internet, can browse and read digital objects created and made available by the editor, cannot use neither editor nor administrator application, every person which visits *dLibra* web pages is treated as a reader.

These are three types of uses commonly used in digital libraries based on *dLibra*. Each has different duties and nature. Administrator takes care of a digital library as a whole, editor cares about the content of the digital library according to limits specified by the administrator, reader uses WWW to access documents (publications) stored by the editor in the digital library.

1.3. Directories

All the digital objects stored in the *dLibra* library are organized in a hierarchical structure of directories. Every directory can contain any number of objects or subdirectories. Directories provide a structural order in a library. In most cases documents will be assigned to directories accordingly to their owner or e.g. to the company department they belong to.

Library structure organized in a structure of directories is visible only in administrator and editor application therefore reader does not see it. Thanks to this approach editor can freely create the structure without any influence on reader view.

1.4. Collections

Directories are only meant to help editor order library publications. Logical structure seen by the reader is kept by the mechanism of library collections. In *dLibra* system there can be any number of collections and their structure is hierarchical. Every collection can contain any number of subcollections. There can be any number of publications that belong to a single collection, and each publication may belong to any number of different collections. Collections are in most cases created for a specific topic (eg. *Educational materials* and its subcollections *Biology*, *Computer science*).

1.5. Publication

Digital object in the *dLibra* system is called publication. In the following sections those two terms will be used exchangeably. Digital object in *dLibra* system (publication) is a unit of information such as an article, report or user's manual. *dLibra* publications consist of files. In the simplest case a publication can contain only one file that conveys all the information (e.g. a PDF, PS or MS Word document). Nevertheless, it is also possible that a single publication will cnsist of many files of different types. The prime example of such a publication is a web page with HTML, JPEG, GIF and other files as its components.

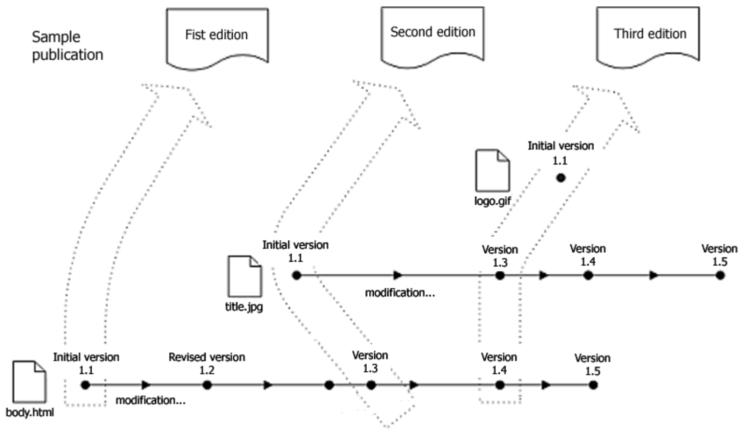
1.5.1. Publication's editions, files and files' versions

Publication's edition is a concrete version of a digital document. Let us assume that an author has created document *A* - this is the first version of the document. The author has sent document to editor which has added it to digital library by creating a publication (when creating publication *dLibra* system creates first edition automatically). Next, the author modified the document and as a result editor has got a new document version - document *A′*. In order to make the new document version (*A′*) avaliable in the digital library the editor should add a new edition to the publication (not new publication!). After a new edition is added publication contains two editions - the first one refers to document *A*, the second one refers to document *A′*. Thanks to this approach users have entire document's “life history” avaliable.

In the context of files an edition is a set of concrete files' versions, where by the file's version we understand a file in a specific time (e.g. file from 17.01.2006 at 07:32 am). It covers with the information from previous paragraph - document *A* is the first version which groups files' versions sent to editor after the document was created. Document *A′* groups files' versions sent to editor after modification. Summing up, the first publication's edition (created automatically when creating publication) groups files' versions of the document *A*, the second edition groups files' versions of the document *A′*. In addition, if the document was composed of many files and the modifications done by the author were not refered to all the document files then the second edition would group files from document *A* (not modified files) and files from document *A′* (modified files).

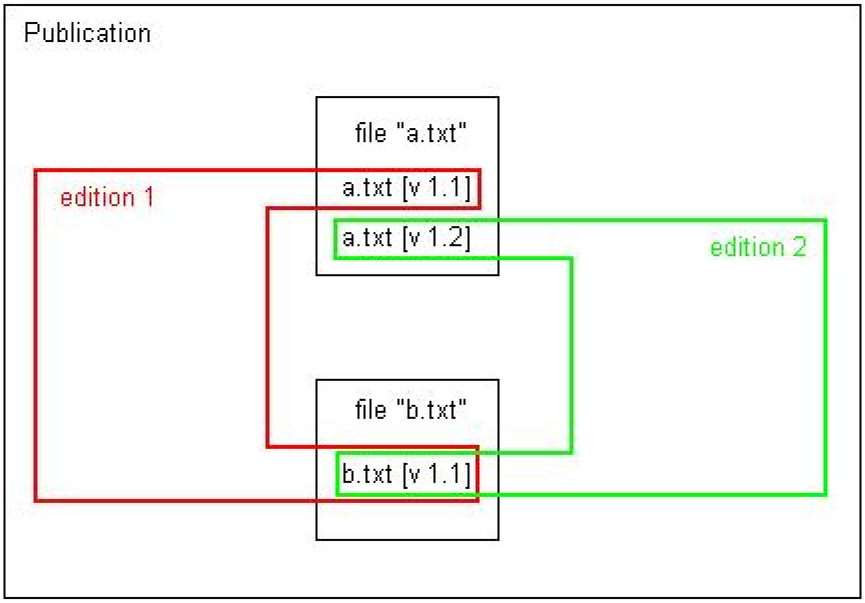
To explain versioning more precisely let us follow an example (illustration Figure 1.1, “File versioning and publication editions”). Let us assume that at the beggining a document is composed of one file - `body.html`. Editor creates a publication in the *dLibra* system by pointing out this file as a content. System creates publication and automatically adds a new edition which groups first version of `body.html` file. After some time the author has modified `body.html` file and added `title.jpg` file. The editor has created second edition which now groups `body.html` file (second version) and `title.jpg` file (first version). After some time the author has modified document again - `body.html` and `title.jpg` file has changed and new `logo.gif` file has been added. The editor has created third edition which groups the newest versions of `body.html` and `title.jpg` file and first version of `logo.gif` file. Summing up, publication is composed of three editions. These editions group files in concrete versions. Publication contains also three files which group concrete files' versions (eg. `title.jpg` file groups version *1.1* and version *1.3*, and file `logo.gif` groups one version *1.1*).

Figure 1.1. File versioning and publication editions



Another example (picture below) presents dependencies between publication, its editions, files and files' versions. The picture below presents a publication which contains two editions and two files. Files contain concrete versions while editions group these versions. WWW user (reader) sees the publication and its two editions. When he wants to see the content of specific edition than *dLibra* system serves files' versions which this edition groups.

Figure 1.2. Struktura publikacji



1.6. Object's properties

Objects in *dLibra* system have various properties. Examples of properties are: object's name, object's metadata, administrative information corresponding to object. Additionally, every property may be multilingual or not, required or optional. Appendix I, *Objects'* *properties in dLibra system* contains information about objects' properties in *dLibra* system.

1.6.1. Multilingual properties

If a property is independent from language then it is specified only once (in one language). If a property is multilingual then is may be specified in any number of languages (used languages are defined by the library administrator). An example of language independent property is directory name. Directories are created by the editor in order to organize documents in hierarchical manner. Directories organization is internal, which means that only editors may see it - WWW reades do not have access to directories. As the directory is an internal object it is not necessary to provide its name in many languages. An example of language dependent property (multilingual property) is collection name. Collection name is presented on WWW pages therefore it is required to have it in many languages (because of different nationality of users which visit WWW pages).

1.6.2. Object's metadata

Some objects in the *dLibra* library can be described by set of attributes called schema. The main scheme in *dLibra* system is Dublin Core 1.1 but there is a possibility to modify it. The values of object attributes are considered while searching the library and are the key element to identify certain resource. Schema is in practice object's metadata. In case of default schema (Dublin Core 1.1) the description consist of information about author, title, publishing date, publisher, etc.

In the *dLibra* system the following elements may have metadata::

- planned publication,
- edition,
- group publication,
- directory

1.7. Other library resources

Apart from the library content (i.e. publications) dLibra manages some other resources. A system of users and groups of users facilitates right management. All rights can be granted on user or group basis --- assigning a user to a group gives him or her all the rights the group has.

In *dLibra* an attribute (which is a part of metadata scheme) is also a resource. Thus, attributes can be created and altered by the library administrators.

Chapter 2. Program installation and startup

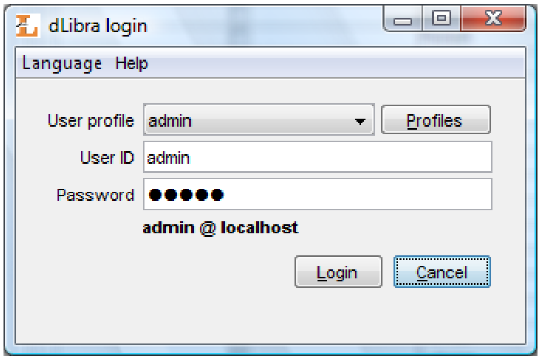
2.1. Installing and running editor's and administrator's application

dLibra GUI application installation is maintained by *dLibra* library administrator. In order to run this application Java 2 Virtual Machine version 6.0 or newer is needed. If the library administrator has installed the application and he or she has created the shortcut (e.g. on the desktop) to *dLibra* GUI application then in order to run the application double click on the shortcut.

If such a shortcut does not exist, in order to run the *dLibra* GUI application:

1. Run web browser (e.g. Internet Explorer in case of Windows operating system)
2. In the address field type the library web address and add the */jnlp* ending. For example, if the library web address were *www.wbc.poznan.pl* then you should type *www.wbc.poznan.pl/jnlp* therefore add to the web address the */jnlp* ending.
3. Approve the web address, e.g. press the **ENTER** key.
4. Wait until needed file will be downloaded and application will be started.
5. If during the application startup secuirity question will appear --- press Always button.
6. On the screen login window should appear (Figure 2.1, “Login dialog”).

Figure 2.1. Login dialog



User profile: admin

User ID: admin

Password: ●●●●●

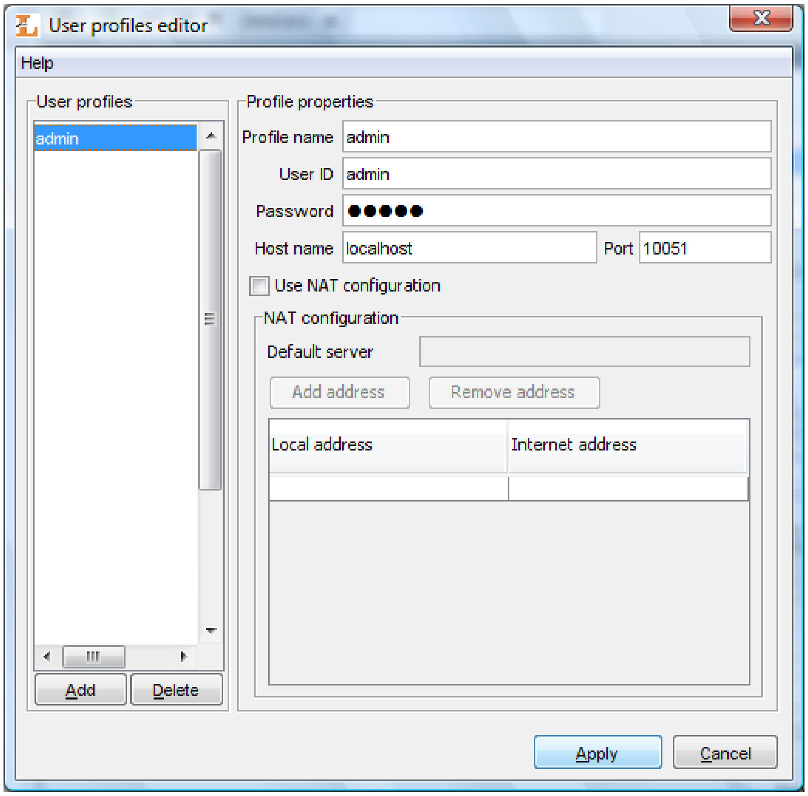
admin @ localhost

2.2. Logging into the *dLibra* system and running editor's or administrator's application

Before any task is performed using *dLibra* GUI applications an authorization process must be carried out. After the *dLibra* GUI application startup login dialog (Figure 2.1, “Login dialog”) will appear on the screen.

The needed authorization data (user id, password) can either be entered directly in User ID and Password fields or stored in a login profile. Login Profiles Editor (Figure 2.2, “Login Profiles Editor”) makes it possible to add new profiles or alter the existing ones.

Figure 2.2. Login Profiles Editor



Profile name: admin
User ID: admin
Password: ●●●●●
Host name: localhost
Port: 10051
□ Use NAT configuration

The default login profile defines the address and port number of the *dLibra* server to be used when authorizing with directly entered user id and password. For the other profiles the data can be defined separately. When leaving an empty password in a login profile the password shall be entered in the Password field of the Login Dialog (Figure 2.1, “Login dialog”) every time the login operation is performed. Additionally profiles may have NAT configuration.

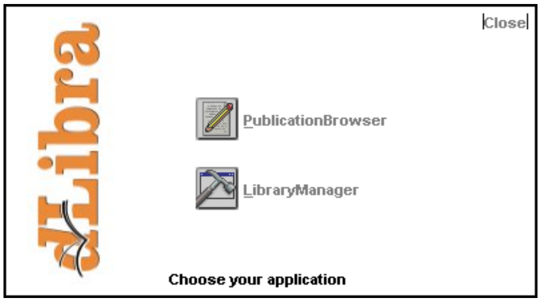
After the successfull login startup application (Figure 2.3, “dLibra startup application”) is shown. It enables the user to run *dLibra* GUI applications. You can choose the application by pressing on its name label:

- *PublicationBrowser* - the editor's application

- *LibraryManager* - the library administrator's application

After application selection loading its components is performed. Information about the component being currently loaded is shown on the bottom of the startup screen.

Figure 2.3. *dLibra* startup application

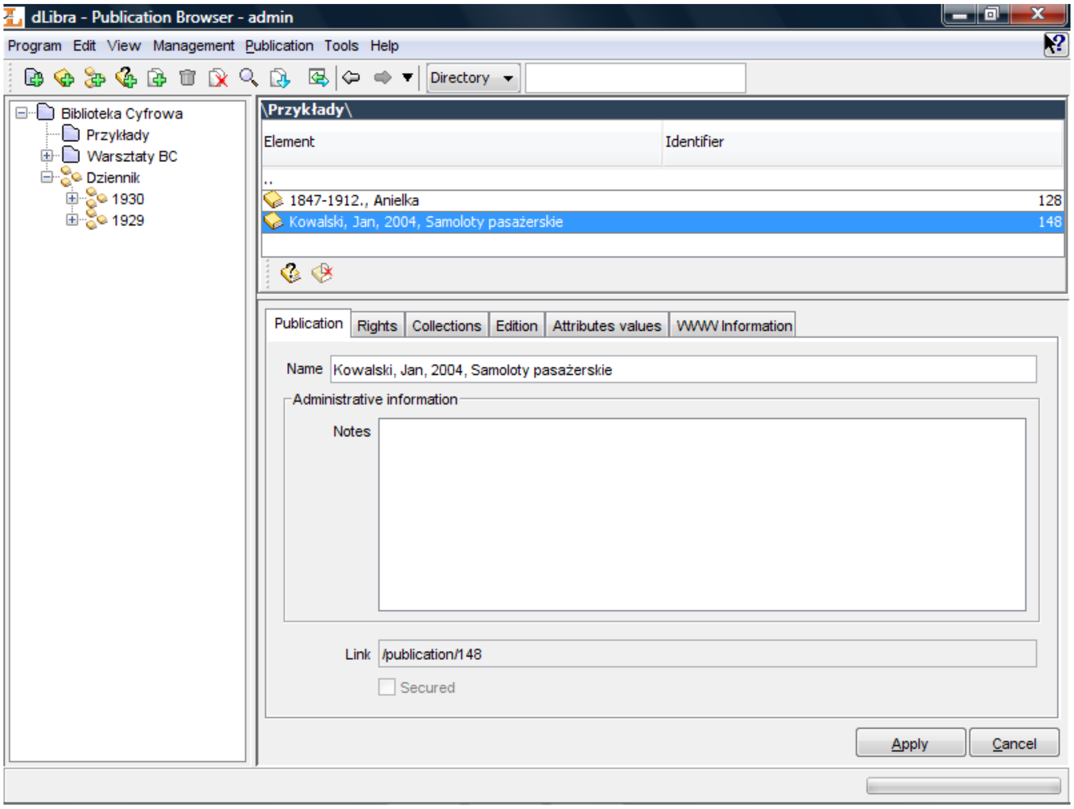


Chapter 3. Editor's Application

This chapter describes the *dLibra* Editor's Application with regard to its basic components.

3.1. Editor's Application - main window

Figure 3.1. Editor's Application main window


dLibra - Publication Browser - admin
Program Edit View Management Publication Tools Help
Directory
Biblioteka Cyfrowa
Przykłady
Warsztaty BC
Dziennik
1930
1929
\Przykłady\
Publication Rights Collections Edition Attributes values WWW Information
Administrative information
Notes
Apply Cancel

Element	Identifier
..	
1847-1912., Anielka	128
Kowalski, Jan, 2004, Samoloty pasażerskie	148

Name: Kowalski, Jan, 2004, Samoloty pasażerskie

Link: /publication/148

□ Secured

The main window of the Editor's Application is divided into five parts:

- Directories tree placed on the left side of editor's application main window. It presents hierarchical structure of the library,
- List of elements placed on top right part of the main window. It presents a list of elements,
- Properties Window (on the right bottom part) which presents properties of an element chosen on the list of elements,
- Main menu placed on the top of the editor's application,
- Toolbar (placed below the main menu) enabling convenient access to most common operations (e.g. creating new publication).

Default editor application view presented above can be configured by the user. User may change components in places where elements list and properties window is placed. In both places user may put either element list or properties window. So it is possible to compare properties of two elements (when properties window is in both places) or manipulate on two lists.

In order to change component where by default elements list is placed (top right corner) press **ALT+F1** - properties window will replace elements list. In order to get back to elements list, press **ALT+F1** again. In order to change component where by default properties window is presented (bottom right corner), press **ALT+F2** - elements list will show up, to get back to properties window, press **ALT+F2** again.

3.1.1. Directories Tree

The directories tree is placed on the left side of the editor's application. Directories allow editors and administrators to order hierarchically elements in the library. Directories are visible only for editors and administrators (it is not visible for WWW users). Directories tree may contain:

- directory node - represents a library directory, may contain subdirectories and group publications,
- group publication node - represents a group publication, which may contain publications.

With every node a context menu is associated that provides quick access to the most common operations performed on a given type of element (e.g. creating a publication). Context menu may be called by clicking the right mouse button on a node.

3.1.2. Elements list

3.1.2.1. Introduction

Elements list allows user to browse the content of the digital library. By default, elements from the main directory are listed on the elements list. You can open an element on the list by doubleclicking on it or by pressing **ENTER** key (when the element is selected). In order to go one level up press **BACKSPACE** key or doubleclick element on the first position - “..” (this elements is not displayed when main directory is opened as there is no higher level than main directory). The path to currently opened element is displayed on the top of the elements list.

When user selects an element on the list properties window (bottom left part of the window) is filled with information about the selected element.

Types of elements possible on the elements list:

- directory node - represents a library directory, may contain subdirectories and publications,
- group publication node - represents a group publication, which may contain publications including group publications.
- planned publication node - represents planned publication and does not contain nodes.

- publication node - represents a publication, in advanced mode contains publication editions nodes and publication files node, whereas in simple mode when publication has only one edition contains files versions of this edition,

- removed content publication node - represents publication which content was removed, contains only publication editions nodes.

- edition node - represents an edition of a publication, contains all versions of publication files that compose the edition,

- publication files node - contains all publication files

- folder node (folder icon the same as in operating system) - represents a folder that groups publication files or publications files versions,

- publication file node (icon depends on the file type) - represents a publication file, contains all versions of the file

- version node (icon depends on the file type) - represents a single version of a publication file

With every node a context menu is associated that provides quick access to the most common operations performed on a given type of element (e.g. adding new files versions). Context menu is called by clicking a right mouse button on a node.

Two toggle buttons are placed on the bottom of the elements list. These buttons allow to hide specific elements. Currenlty, it is possible to hide planned publications and publications with removed content.

3.1.2.2. Description of selected elements

- Publication

  The structure of publication is described here. Publication in the *dLibra* system may be secured. Publication can be secured only If web application is able to secure it. By default web application is able to secure HTML (up to version 3.2 - higher versions may be incorrectly displayed), DjVu and PDF publications. In orde to secure other types of documents web application extension has to be prepared. Secuirity mechanism disallows reader to copy, save or print publication content.

3.1.3. Properties Window

“Properties Window” consist of a set of tabs (the number depends on the chosen element's type). Each tab contains properties editor which allows user to edit some properties of the element selected on the elements list. Pressing the Apply button will store the information from tabs on the server. Pressing the Cancel button or selecting another element on the elements list will discard all changes made since the last Apply operation.

3.2. Editors of object's properties

Properties of the element selected on elements list are displayed on the properties window. Properties window is composed of several tabs (the number of tabs depends on the type of selected element). Each tab contains an editor which allows user to modify defined set of element's properties. This section describes the most important properties editors.

3.2.1. General information editor

General information editor allows user to view/modify element's main properties, for example directory name for directory element. General information editor is usually the first tab in the properties window. The name of the tab is the name of edited element's type.

The following list shows properties for specific library objects:

- *Library directory*:
  - *Name* - defines name of a directory identifying it in *dLibra* system; directory name is language independent,
  - *Notes* - administrative notes for a directory; notes content is language independent,
- *Library collection*:
  - *Name* - defines name of a collection identifying it in *dLibra* system and visible in WWW application; collection name is language dependent and should be defined for each language separately,
  - *Description* - defines description and contains basic information about collection, visible also in WWW application; collection description is language dependent and should be defined for each language separately,
  - *Notes* - administrative notes for a collection; notes content is language independent,
- *Publication (planned, normal and without content)*:
  - *Name* - defines name of a publication identifying it in *dLibra* system; publication name is language independent and is visible only in Editor's Application and Library Manager,
  - *Notes* - administrative notes for a publication; notes content is language independent,
- *Edition*:
  - *Name* - defines name for an edition which is language independent; edition name is visible in WWW application,
  - *Notes* - administrative notes for an edition; notes content is language independent,
  - *Published* - indicates whether this edition is available on the web pages or not

3.2.2. Bibliographic description editor

3.2.2.1. Attributes values

dLibra system allows editor to describe stored elements. Description of a specific element is defined by values assigned to different attributes. Set of attributes which is used to describe elements is called metadata schema. Values which describe element are called metadata. Metadata schema can be composed of attributes such as *Author*, *Title*, *Date*, etc. Examplary values describing specific element are: *1920* for attribute *Date* or *John Smith* and *Kate Smith* for attribute *Author* (it is possible to assign more than one value to specific attribute). This section describes the bibliographic description editor and the way metadata assigned to *directory*, *group publication* or *edition* influence on other elements in the system.

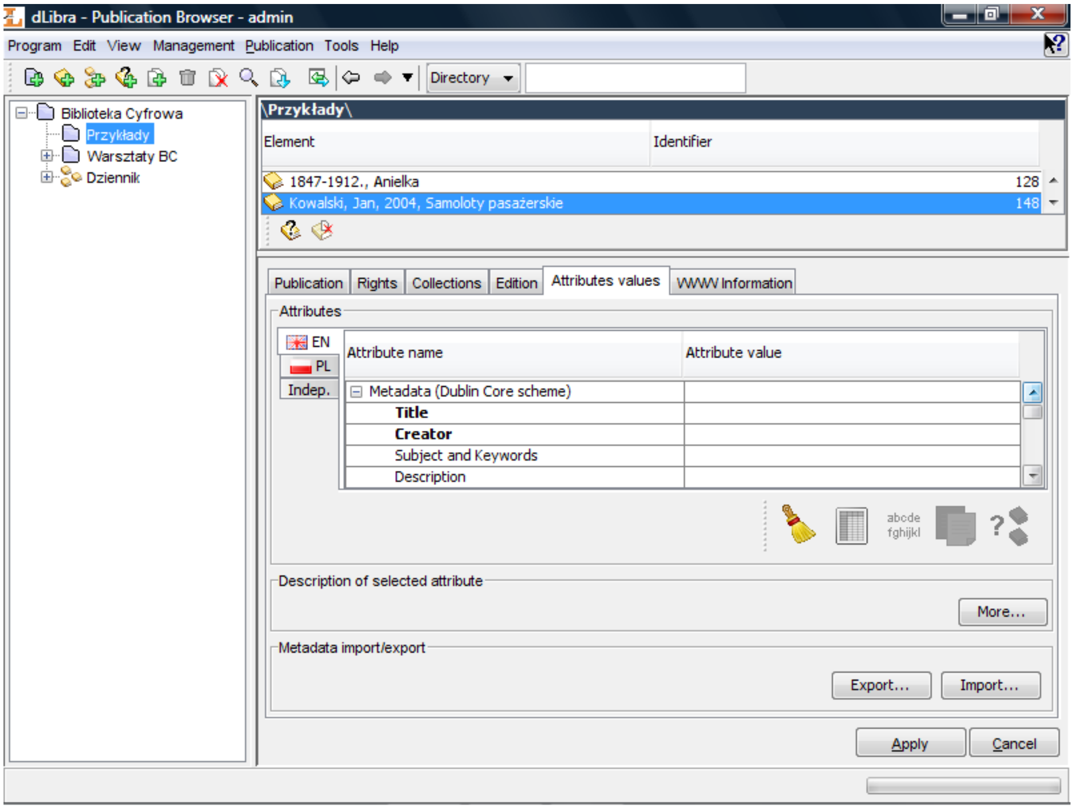
Because of the fact that each element which may have metadata is described using the same metadata schema it is possible to create a dictionary of all values used in a system for specific attribute in specific language. Additionally each dictionary supports idea of synonyms which is presented in attribute's values dictionary section. Describing library resources with metadata is based on choosing a set of values from dictionaries of some attributes. If a value doesn't exist in a dictionary, it is possible to add it using Publication Browser.

3.2.2.2. Attributes' Values Editor

> **Note**
>
> Values in bibliographic description editor should not contain HTML tags (e.g. `<B>`). These tags will be displayed as part of the value - these tags will not be interpreted as HTML code by the web browser. For example, if the editor will type `<B>Example</B>` value then on the web page reader will not see bold `Example` value but the exact typed value which is `<B>Example</B>`.

The Attributes' Values Editor (Figure 3.2, "Attributes' values editor") allows user to modify bibliographic description of edition, group publication or directory. Bibliographic description of an edition, group and planned publication is essential for searching mechanism in the reader's application (WWW pages). Description of a group publication is inherited by all the group publications and editions placed lower in the inheritance hierarchy. Directory description is a default bibliographic descritpion for all newly created elements in this directory.

Figure 3.2. Attributes' values editor



Element	Identifier
1847-1912., Anielka	128
Kowalski, Jan, 2004, Samoloty pasażerskie	148

Attribute name	Attribute value
Metadata (Dublin Core scheme)	
Title	
Creator	
Subject and Keywords	
Description	

Attributes' Values Editor is available on Attributes values tab. It allows user to describe a library element with a set of attributes available in system. Bibliographic description is presented in a table with two columns. Attributes tree is placed in the left column, values assigned to attributes are in the right column. Every attribute may have many values. Every value is presented in separate row starting from the row with attribute name. If an attribute has no values assigned then the field on the right side of the attribute name is empty.

In order to add value to an attribute which has no value assigned:

1. Choose an empty field on the rights side of attribute's name (e.g. clik left mouse button on it) to which you want to add a value,

2. Enter the value you want to add,

3. Approve the operation by pressing the **ENTER** key.

In order to add value to an attribute which have at least one value assigned:

1. Choose a field with value assigned to the attribute,

2. Press the **TAB** key - a new row will be added (below selected row)

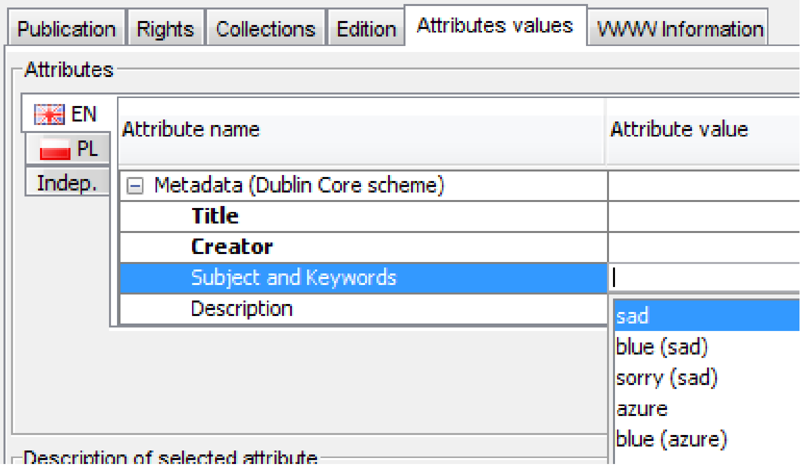
3. Enter the value you want to add,

4. Approve the operation by pressing the **ENTER** key.

While entering attribute's value it is possible to press the **TAB** key. It will add the current value to attribute's values list and create a new row in which next value can be added.

Autocomplete mechanism will be started if current value matches any value of the attribute's dictionary. The autocomplete mechanism shows the list of attribute's values (see Figure 3.3, “Attribute's values list shown by the autocomplete mechanism”) which start with current value. The user may select the value from the autocomplete list (e.g. by double clicinkg on its name).

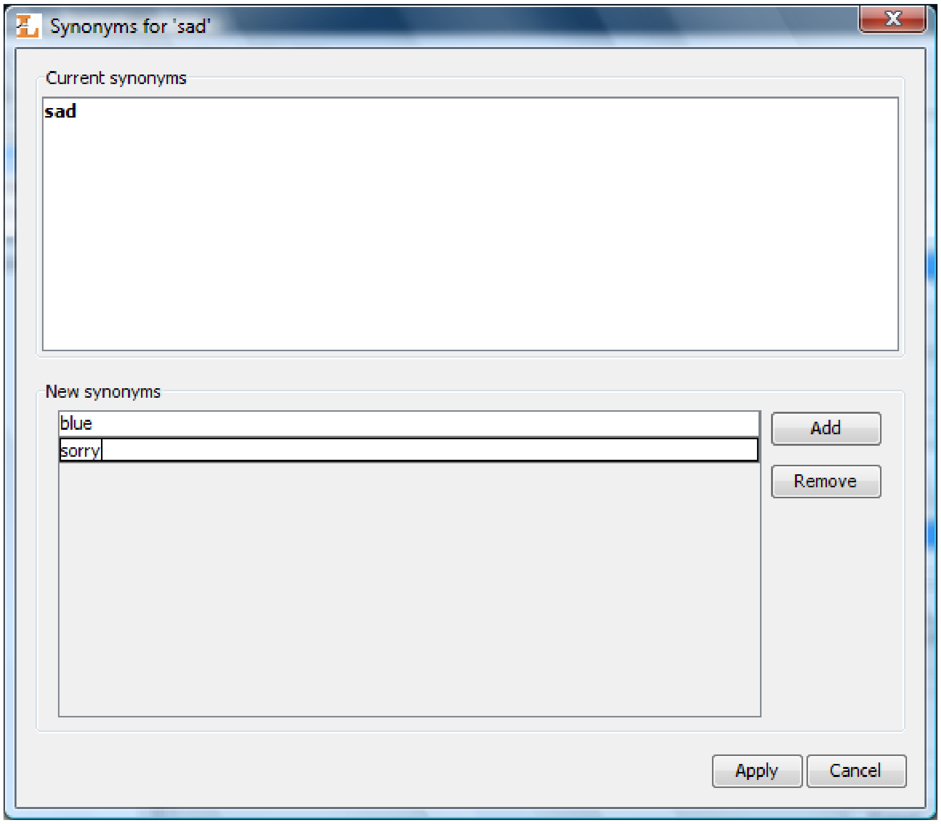
Figure 3.3. Attribute's values list shown by the autocomplete mechanism



Attribute's value can be removed from the attribute's values list. In order to do it select a value on the list and press **DELETE** key (on keyboard). The value will be removed from the bibliographic description (if there was only one value - the field with it will be cleared, if there was more than one value - the row with selected value will be removed).

As mentioned earlier metadata are supported by synonyms dictionary mechanism (detail can be found here). The metadata editor allows user to provide synonyms for value selected on the metadata table. In order to add synonyms to selected value press the combination of **Ctrl** and **S** keys (**Ctrl**-**S**) on keyboard. The synonyms dialog for selected value (Figure 3.4, “Synonyms dialog”) will be opened.

Figure 3.4. Synonyms dialog



Synonyms for 'sad'

Current synonyms

sad

New synonyms

blue

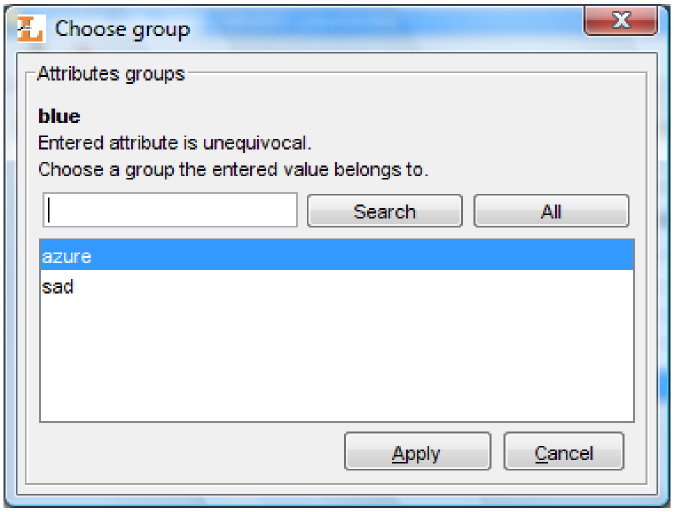
sorry

The synonyms dialog presents current synonyms (Current synonym panel) of the value and allows editor to add new synonyms on the New synonyms panel. In order to add first synonym just type the value to the first position of the new synonyms list (New synonyms panel). To add next synonym press Add button and type the value to newly created position on the synonyms list. To remove value from new synonyms list select it on the list and press Delete button. To apply all changes press Apply button, to cancel chages press Cancel button.

Attention! In order to save attributes' values (bibliographic description) press Apply (the bottom right corner of the “Properties Window”).

If equivocal value is present in bibliographic description user must choose the meaning of the value. In order to do this application pops up choose group window (see Figure 3.5, “Group selection for a value”) so the user could choose the group (the meaning) that the equivocal value belongs to. An example: Let us assume that the bibliographic description contains the *blue* value and the *dLibra* system have to groups of values which contain *blue* - *sad* group and *divine* group. In such case the user have to decide either the *blue* is a color (*divine* group) or it is a state of mind (*sad* group).

Figure 3.5. Group selection for a value



Choose group

Attributes groups

blue

Entered attribute is unequivocal.

Choose a group the entered value belongs to.

The bibliographic description is language dependent which means that the user may specify metadata in a number of languages. To view/modify bibliographic description in specific language user have to select appropriate tab. The name of the tab is two-letter abbreviation of the language name and it is pictured by the icon representing the language.

A set of buttons which allow editor to perform additional operation on the metadata are available under the bibliographic description table. The buttons are (respectively, from the left):

- - clears metadata editor (removes all added values). All values added to element's metadata are removed.

- - displays metadata in a separate window (it facilitates metadata browsing when there are long values in the description).

- abcde fghijkl - displays in the metadata table all values which are inherited from overriding group publications (values are displayed in red, it is not possible to modify inherited values, it is not possible to add values to the attribute which inherits values). This button is disabled if the element cannot inherit metadata (is not placed in a group publication).

- - copies all inherited values to metadata of the element (inherited values are directly added to the metadata of edited element). This button is disabled if the element cannot inherit metadata (is not placed in a group publication).

- ?, ≠, ~ or = - basis on provided metadata, searches for potential duplicates in the DLF service (Digital Libraries Federation of the PIONIER Network - http://fbc.pionier.net.pl). Values from attributes marked as bold are used as search terms. When the editor fills up the metadata, the application automatically checks whether the potential duplicates exist and basis on the result it updates the icon of the button as follows:

- ? - means that values in the metadata are not sufficient to perform search with satisfactory acuracy and state whether potential duplicates exist. The icon is also present when the application is performing the search operation or the DLF service is not available.

- ≠ - means that potential duplicates were not found.

- ~ - means that potential duplicates have been found, but the search query was not complete (because some attributes which are taken into account in the search operation have no values assigned) therefore the result is approximate.

- = - means that potential duplicates have been found and the search query was complete (because all attributes which are taken into account in the search operation have assigned values) therefore it is highly possible that found elements are actual duplicates.

The button is enabled only when creating new elements (e.g. in new publication wizard).

3.2.2.3. Independent metadata values

Let us assume that in our library we provide bibliographic description in two languages (English and Polish). In order to provide an author of a specific publication the editor has to put this value in both languages. In order to modify this value the editor has to modify it in both languages (separatelly for each language). In such situation the author value is independent, which means that it concerns two languages (the same value should be provided for each language). In order to simplify creation of bibliographic description in such situations special independent values language has been added to dLibra system (marked as “Indep.” on bibliographic description editor tab). This language is created only for editors to make their work easier (bibliographic description in independent values language is not visible on WWW pages). Independent values are automatically added to bibliographic description of other languages when presenting these on WWW pages.

For example, let us assume that in editor and administrator application we have typed the value of the “Title” attribute in independent values language for edition *A*. On edition *A* WWW page the user sees this value in every language. The bibliographic description presented in concrete language (e.g. Polish or English) is filled with independent values - all independent values are added to bibliographic description of a language selected on WWW page. Please notice that values from universal language are *added* (not replaced) to values of other languages.

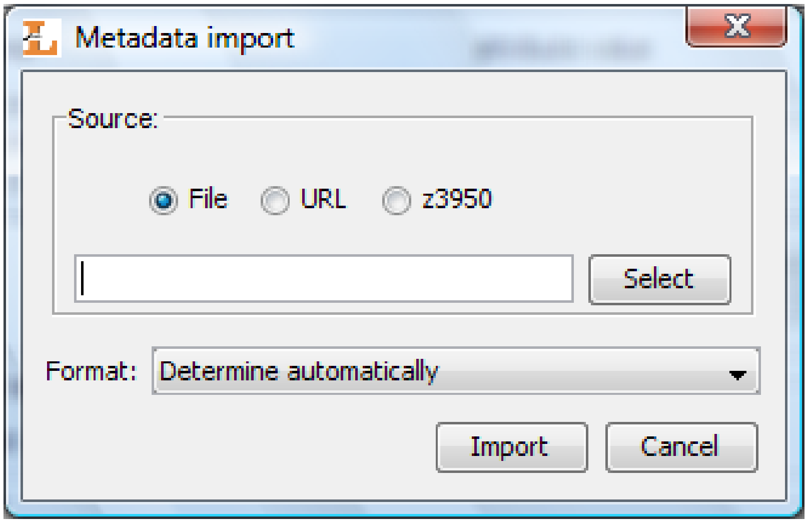
3.2.2.4. Metadata import

Attributes' values editor allows bibliographic description to be imported from external file which structure is in specific format. Digital library administrator may add so called extensions to application in order to enable metadata import in specific format. By default extensions for the following formats are added:

- MARC 21 communication format (details can be found in application configuration and Appendix C, *MARC metadata import*).
- XML format (default configuration allows to import metadata from RDF and MASTER format, details can be found in application configuration and ???).
- Bibtex format (details can be found in application configuration and Appendix F, *BibTeX metadata import*).

In order to import metadata from external resource, press Import... button on the Import/Export metadata panel. Metadata import dialog (Figure 3.6, “Metadata import dialog”) will show up.

Figure 3.6. Metadata import dialog



Metadata import

Source:

● File ○ URL ○ z3950

Format: Determine automatically

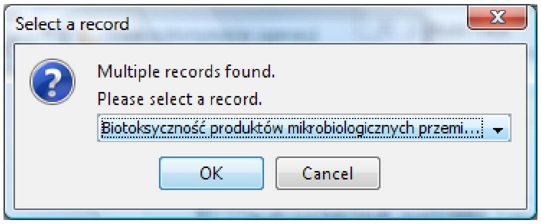
The metadata import dialog allows user to specify the source of the bibliographic description and the format of the source file. Metadata source may be specified by selecting a file on a computer drive, providing a URL adress for a metadata file or selecting a file using extension. In case of the URL option the editor may drag a link from a web page and drop it on the text field where URL should be provided - link will be automatically pasted there. By default editor's application has the following extensions installed for metadata file selection:

- Z39.50 extension - allows editor to select metadata from Z39.50 server (details can be found in appendix concerning Z39.50 extension).

To import metadata from the selected source press Import button. When Automatically detect option (on the Format list) is selected, application tries to detect source format and import the metadata automatically. Otherwise the editor has to select approriate format of the metadata file.

If the file to import contains more that one set of metadata the application will ask the user to choose one. (Figure 3.7, “Choose metadata set”).

Figure 3.7. Choose metadata set



Select a record

Multiple records found.
Please select a record.

Biotoksyczność produktów mikrobiologicznych przemi...

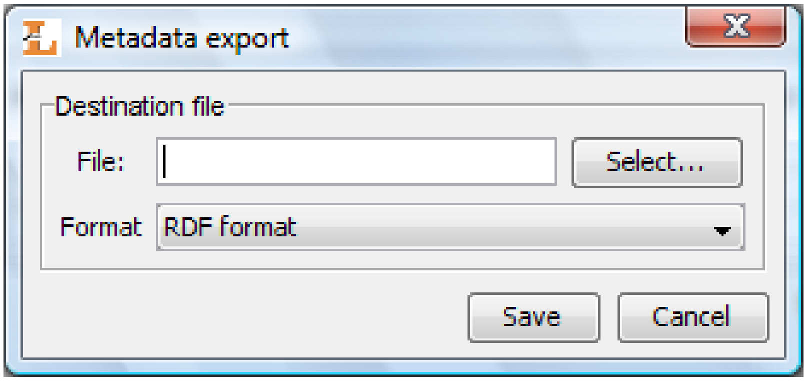
3.2.2.5. Metadata export

Attributes' values editor allows bibliographic description to be exported to external file which will be saved in specific format. Digital library administrator may add so called extensions to application in order to enable metadata export in specific format. By default extensions for the following formats are added:

- RDF format (see application configuration and Appendix D, *Export to RDF format* for details on conversion configuration).

In order to export metadata to a file, press Export... button. Metadata (Figure 3.8, “Metadata export dialog”) export dialog will show up.

Figure 3.8. Metadata export dialog



Metadata export

Destination file

File:

Format RDF format

The metadata export dialog allows user to specify the destination file and the format of the destination file. Press Export button to export metadata.

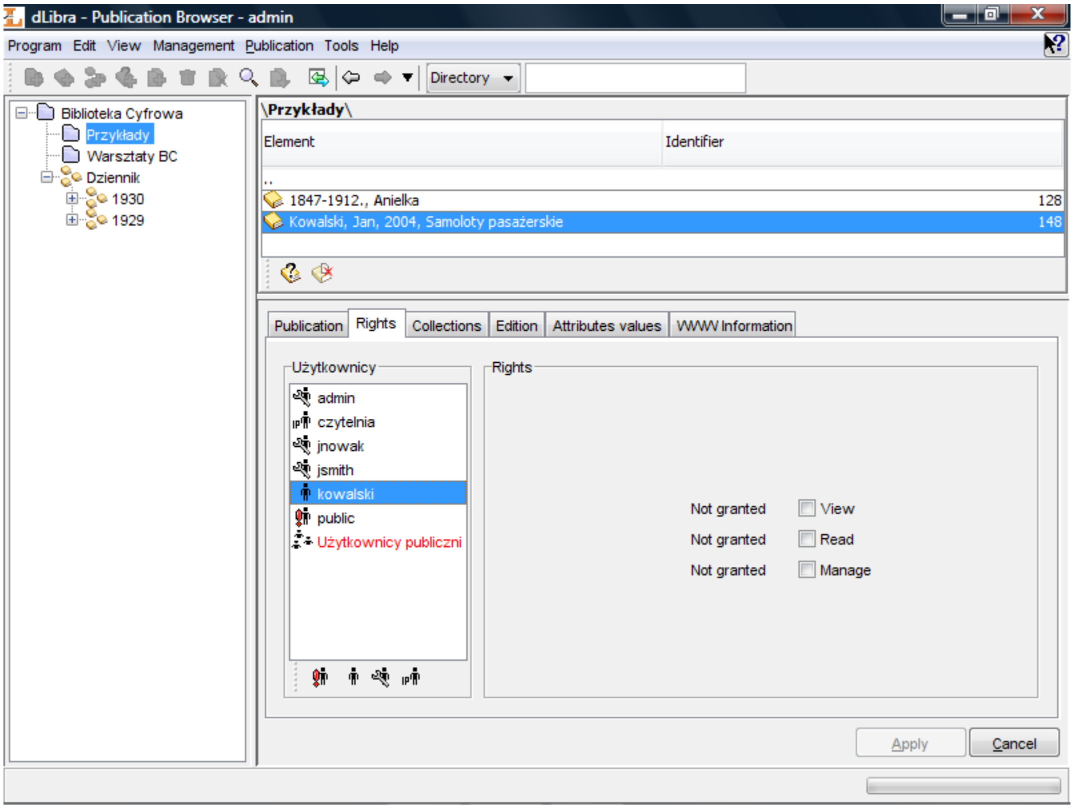
3.2.3. Rights editor

The Right Editor (Figure 3.9, “Right Editor - publication rights”) enables user to alter users access rights to publication.

There are three rights which can be granted to a user (or group of users) for selected publication:

- *View* - The permission to read all published editions of a publication.
- *Read* - The permission to read all editions of a publication.
- *Manage* - The permission to manage the publication (e.g. to create a new edition or grant access rights).

Figure 3.9. Right Editor - publication rights



Element	Identifier
..	
1847-1912., Anielka	128
Kowalski, Jan, 2004, Samoloty pasażerskie	148

Not granted □ View
Not granted □ Read
Not granted □ Manage

Every right can have one of five states:

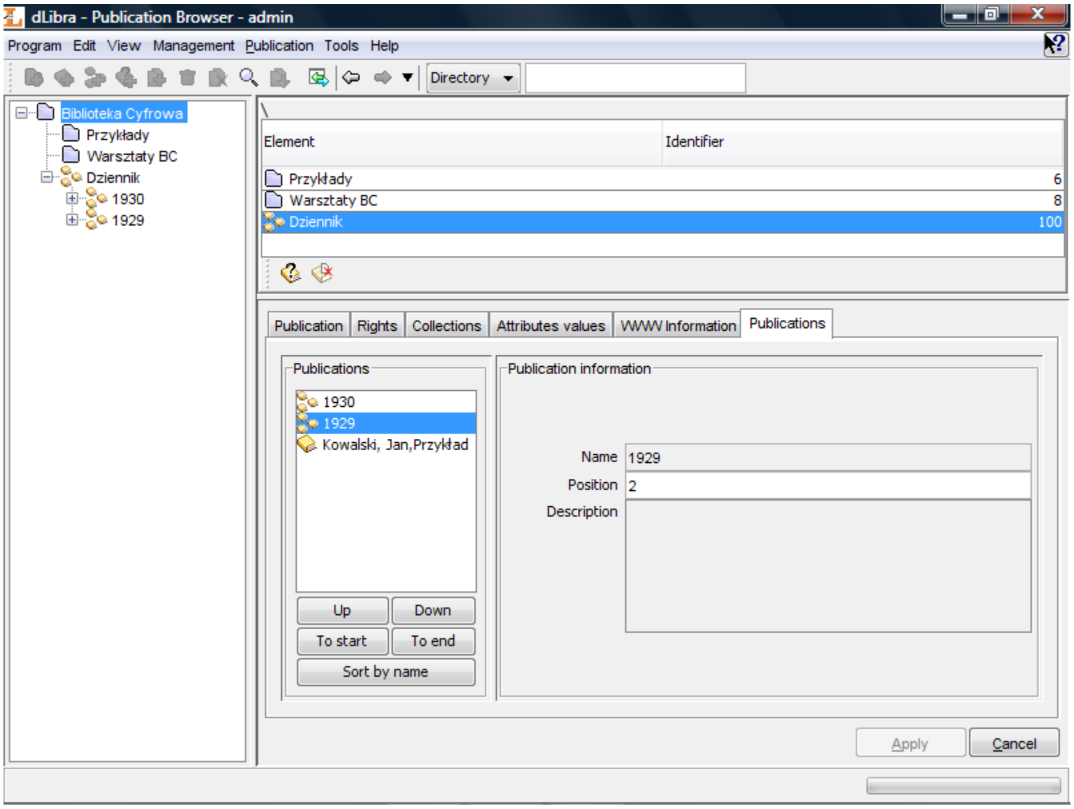
- *Granted* - The permission is granted directly to the user.
- *From group* - The user is a member of a group that has the right granted.
- *Inherited* - The permission is granted to some of the parent objects (e.g. a parent directory).
- *Implied* - The permission is granted because of another right granted (e.g. a user who has a Manage right for a publication has also an implied Read right to the publication).
- *Not granted* - The right is not granted at all.

Rights to publication can be also assigned in administrator application (Section 4.3.5, “Publication-level permissions”).

3.2.4. Publication position editor

Group publication may contain other publications which are ordered by an editor. Publications are displayed on the WWW according to the order specified by an editor. By default, newly added publication appears on the last position. It is possible to manipulate publication position using publication position editor (Figure 3.10, “Publication position editor - publication positioning”).

Figure 3.10. Publication position editor - publication positioning



dLibra - Publication Browser - admin
Program Edit View Management Publication Tools Help
Directory
Biblioteka Cyfrowa
Przykłady
Warsztaty BC
Dziennik
1930
1929
\
Publication Rights Collections Attributes values WWW Information Publications
Publications
Publication information
Up
Down
To start
To end
Sort by name
Apply
Cancel

Element	Identifier
Przykłady	6
Warsztaty BC	8
Dziennik	100

Publications:
- 1930
- 1929
- Kowalski, Jan,Przykład

Name: 1929
Position: 2
Description:

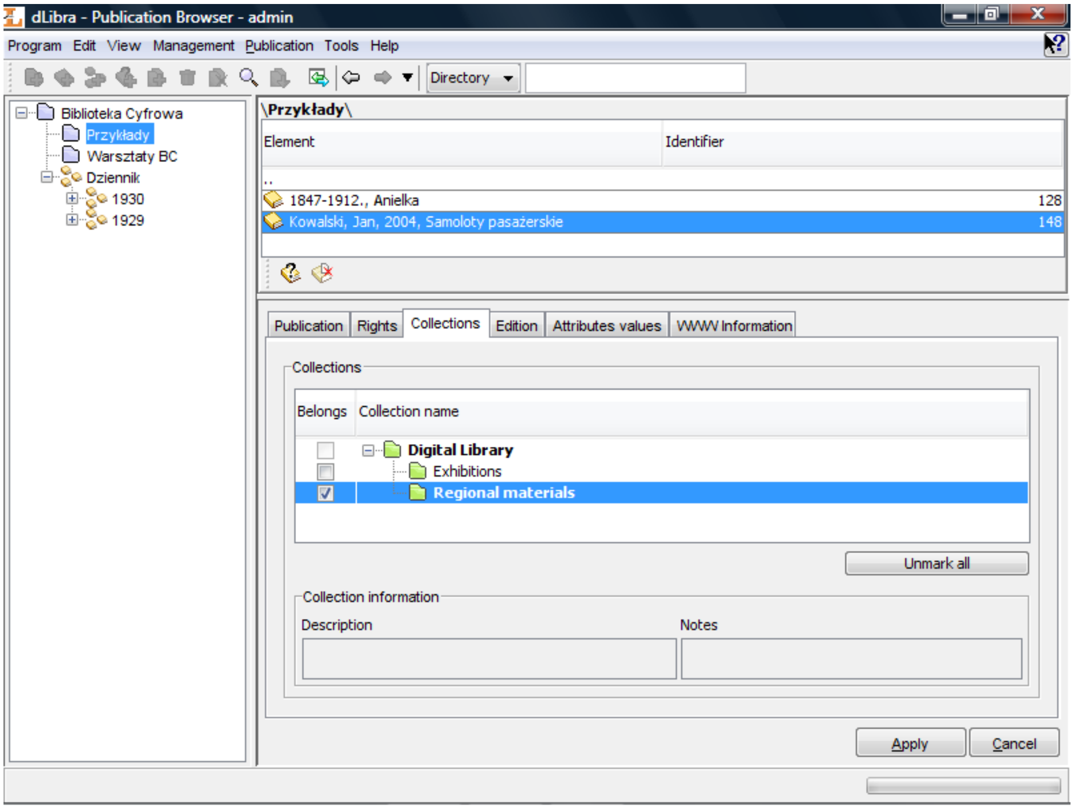
Using the Up, Down, To start or To end buttons on the Publications panel user may change position of selected publication. Basic information about selected publication are presented on the Publication information panel.

3.2.5. Publication to collections assignment editor

Collections are grouped in a library in hierarchical structure. Each collection may contain any number of subcollections. On every level of this structure a collection may be assigned a publication. Any number of publications may belong to a single collection. *If publication belongs to a collection it implies that it also belongs to all its supercollections on the path to main library collection.*

Collections editor allows user to assign publication to a number of collections. When publication or group publication is selected on the elements list the collection editor is available on the Collections tab.

Figure 3.11. Adding publication to collections



Element	Identifier
..	
1847-1912., Anielka	128
Kowalski, Jan, 2004, Samoloty pasażerskie	148

Belongs	Collection name
☐	**Digital Library**
☐	Exhibitions
☑	**Regional materials**

To assign a collection to selected on the elements list publication check appropriate check box. To remove the assignment deselect appropriate check box. Check boxes are placed on the left side of the collection's name.

If the collection icon is green (🟩) it means that currently logged user has rights to manage this collection and may change the assignment of a publication. Red icon (🟥) indicates that user do not have appropriate rights to change the assignment of the publication in a collection.

If a collection name is marked with a bold font it means that selected publication will belong to this collection (directly or indirectly). When collection is checked all its super-collection are also marked with a bold font. It means that the selected publication is indirectly assigned to all supercollections and directly assigned to checked collection.

All the changes must be applied using the Apply button. Choosing different element or clicking the Cancel button discards all changes since the last apply action.

3.3. Typical editor's tasks

This section describes typical editor's tasks. Full list of operations which may be performed on specific objects can be found in Appendix J, *List of operations concerning objects of the dLibra system*.

3.3.1. Creating directory

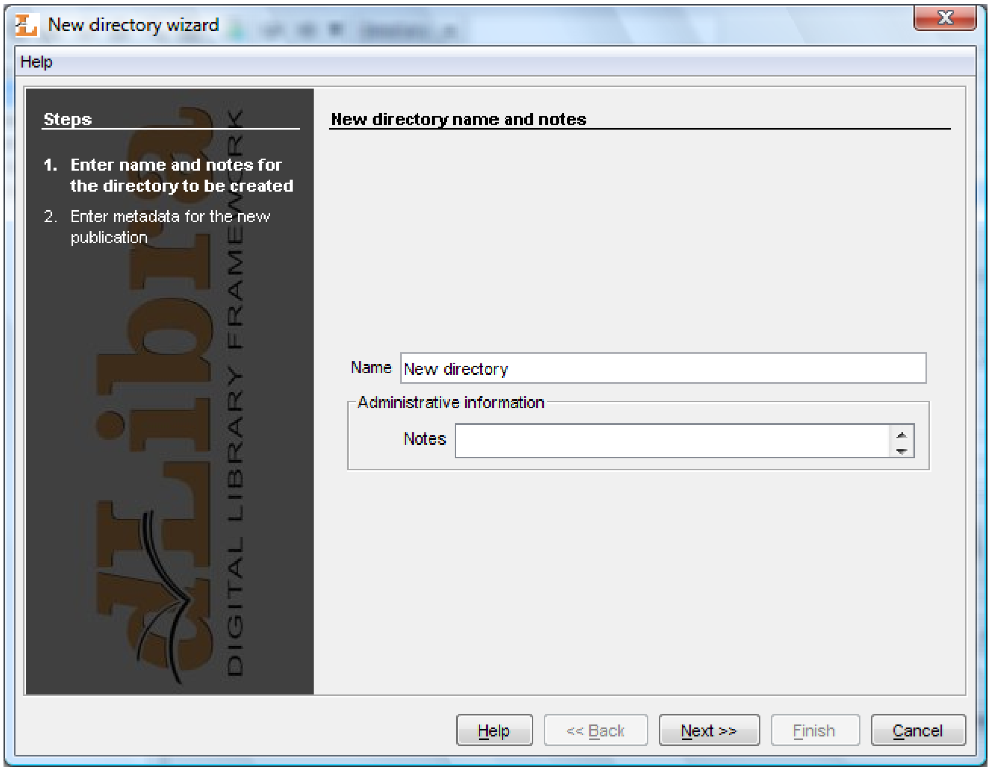
To create a new directory:

1. On the elements list select destination directory (parent directory of new directory) .

2. From the popup menu select New directory function.... Alternatively, the toolbar button or Management main menu option can be used.

3. In the first step of the New Directory Wizard enter name and description for the new directory. Press Next button to proceed to the next step.

Figure 3.12. New directory wizard - general information



Steps

1. Enter name and notes for the directory to be created
2. Enter metadata for the new publication

New directory name and notes

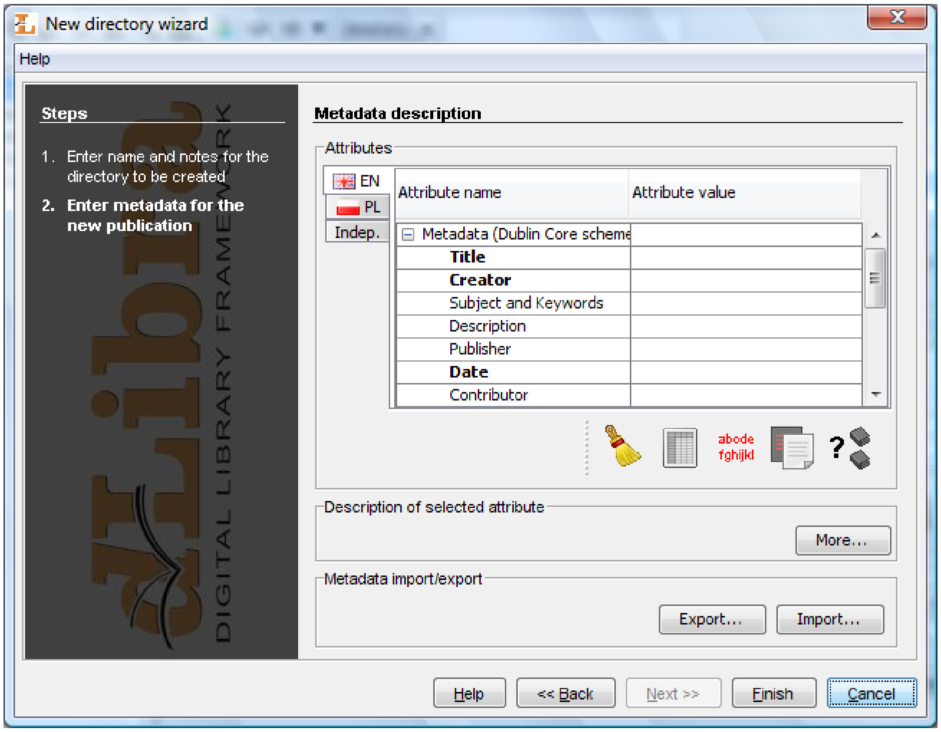
Name: New directory

Administrative information

Notes:

4. The second step allows editor to provide the metadata for new directory (Figure 3.13, “New directory wizard - metadata”). Directory metadata state default bibliographic description of any new element created in it - this description is automatically added to appropriate step in new element wizard and can be modified there. The metadata editor is described in details here.

Figure 3.13. New directory wizard - metadata



3.3.2. Creating a new group publication

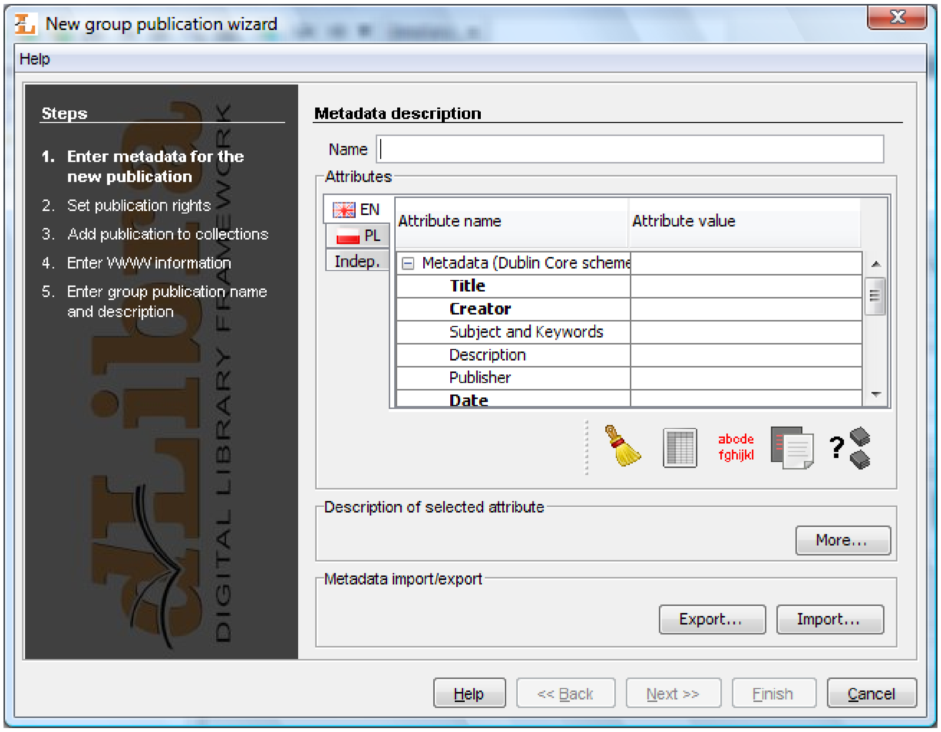
To create a new group publication:

1. On the elements list select the directory or group publication the new group publication should belong to.

2. From the popup menu select the New publication... function. Alternatively, the toolbar or Management main menu option can be used. New group publication wizard will start.

3. In the first step (Figure 3.14, “New group publication wizard - metadata”) provide metadata for the publication. See bibliographic description editor section for details. Provided metadata are inherited by all publications created inside this newly created group publication. In order to proceed to the next step publication name has to be given (Name field). By default the name is created basis on metadata by combining the title, author and publishing date (see configuration to change default settings). For example, if the title is “Stories”, author is “John Smith” and the publishing date is “2004” then the name will be “John Smith, 2004, Stories” - first author, then publishing date and title. Whenever user changes the metadata the name is updated. It is also possible to enter different name or modify default name. Press the Next button to move to the next step.

Figure 3.14. New group publication wizard - metadata



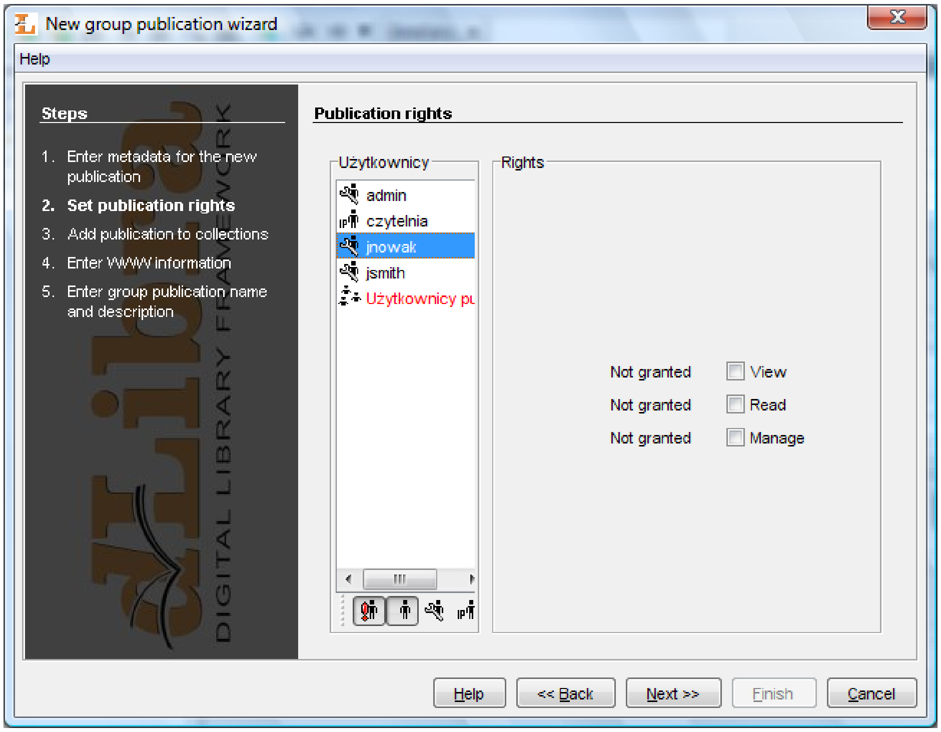
Steps

1. **Enter metadata for the new publication**
2. Set publication rights
3. Add publication to collections
4. Enter WWW information
5. Enter group publication name and description

Attribute name	Attribute value
⊟ Metadata (Dublin Core scheme	
Title	
Creator	
Subject and Keywords	
Description	
Publisher	
Date	

4. In the second step (Figure 3.15, “New group publication wizard - rights”) specify users rights to publication. See rights editor section for details. Rights are inherited by all publications created inside this newly created group publication. Press the Next button to move to the next step.

Figure 3.15. New group publication wizard - rights



Steps

1. Enter metadata for the new publication
2. **Set publication rights**
3. Add publication to collections
4. Enter WWW information
5. Enter group publication name and description

Publication rights

Użytkownicy

- admin
- czytelnia
- jnowak
- jsmith
- Użytkownicy pu

Rights

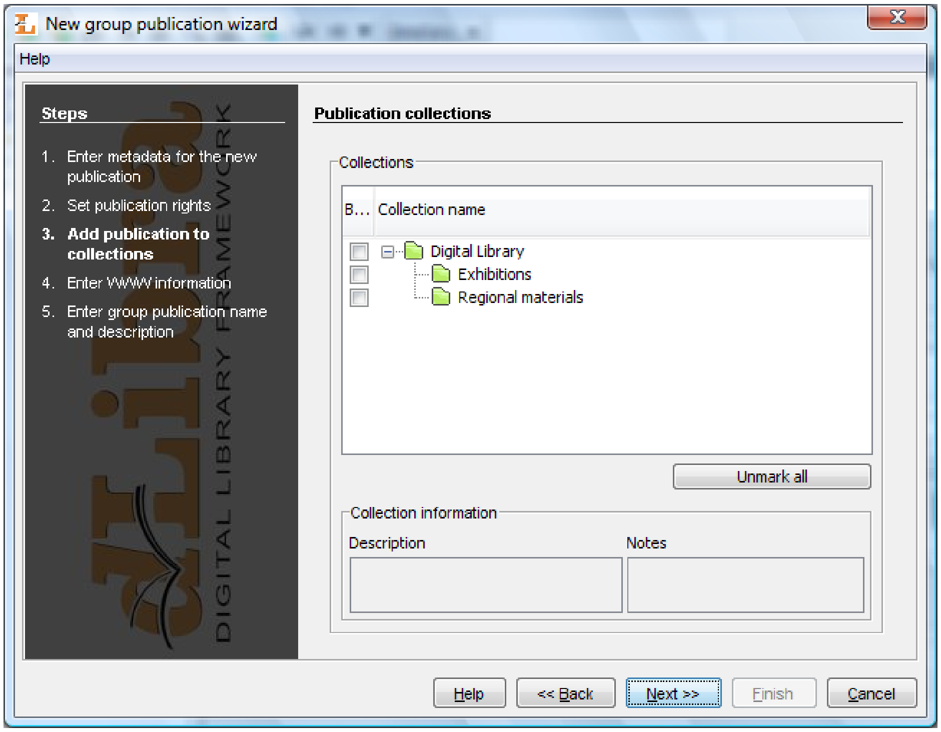
Not granted □ View

Not granted □ Read

Not granted □ Manage

5. In this step (Figure 3.16, "New group publication wizard - collections") specify collections assigned to publication. See collections assignment editor section for details. Collections are inherited by all publications created inside this newly created group publication. Press the Next button to move to the next step.

Figure 3.16. New group publication wizard - collections

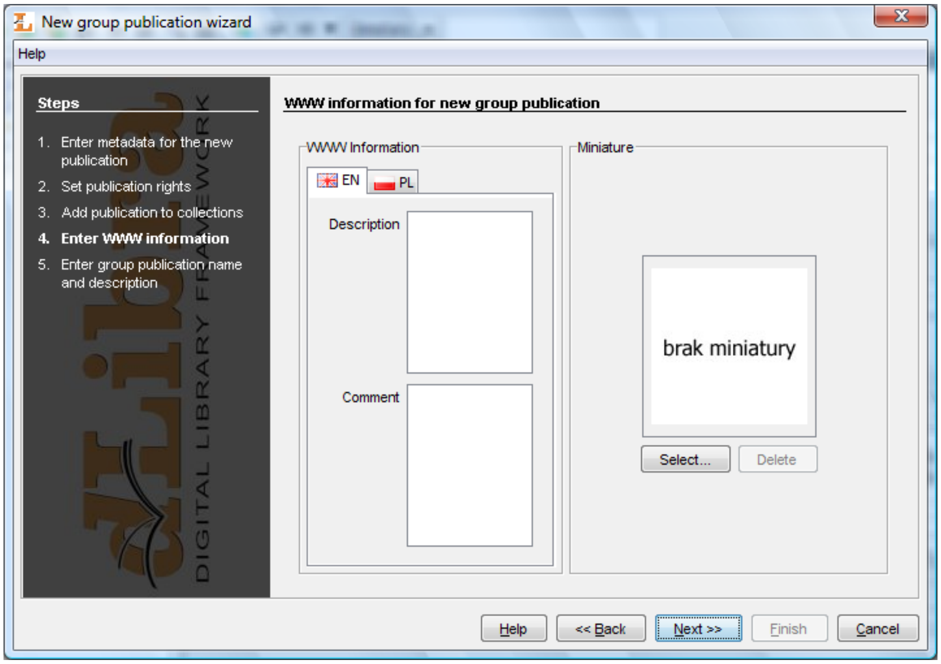


1. Enter metadata for the new publication
2. Set publication rights
3. **Add publication to collections**
4. Enter WWW information
5. Enter group publication name and description

- □ Digital Library
  - □ Exhibitions
  - □ Regional materials

6. In this step (Figure 3.17, “New group publication wizard - WWW information”) specify WWW information: Description which is presented on search results list, Comment and Miniature available on publication information page. Press the Next button to move to the next step.

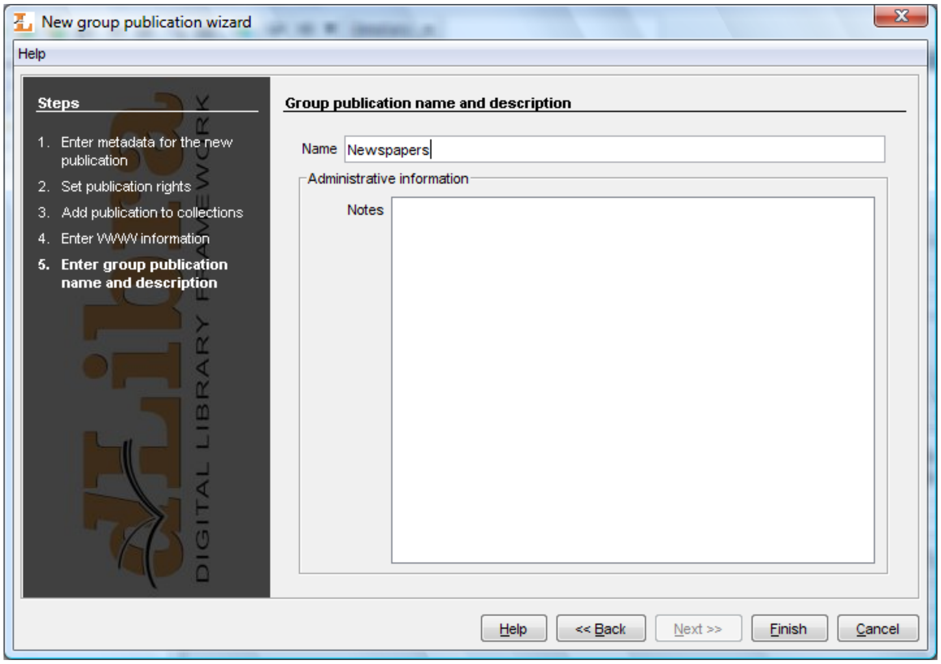
Figure 3.17. New group publication wizard - WWW information



1. Enter metadata for the new publication
2. Set publication rights
3. Add publication to collections
4. **Enter WWW information**
5. Enter group publication name and description

7. In the last step (Figure 3.18, “New group publication wizard - general information”) verify name and enter notes if necessarry. Click Finish to create new group publication.

Figure 3.18. New group publication wizard - general information



Steps

1. Enter metadata for the new publication
2. Set publication rights
3. Add publication to collections
4. Enter WWW information
5. **Enter group publication name and description**

Group publication name and description

Name: Newspapers

Administrative information

Notes:

3.3.3. Creating a new publication

> **Note**
>
> All the publication files have to be placed on a local drive before creating publication in the *dLibra* system.

To create a new publication:

1. On the elements list select destination directory or destination group publication,
2. From the popup menu select the New publication.. option. Alternatively, the toolbar button or Management main menu option can be used,
3. In the first step of the New Publication Wizard (Figure 3.19, “New Publication Wizard - main publication file page”) select the main publication file. If the publication content is in one file, for example PDF or MS Word document, choose this file. If the publication is composed of many files (e.g. HTML files) choose top level file (in case of HTML usually `index.html`).

   If the main publication file is an HTML or DJVU file the wizard will try to find all publication files and verify consistency of the files. If there are any problems with files' consistency Warnings... button will be activated so the user can press this button to see warnings. It is strongly recommended that only complete publications (without missing files) are put into the library. However, using the Ignore inconsistency option it is possible to proceed with an incomplete/inconsistent publication.

All files that the publication consists of will be added to publication files table. There is no way to uncheck them, they all will be sent to the server.

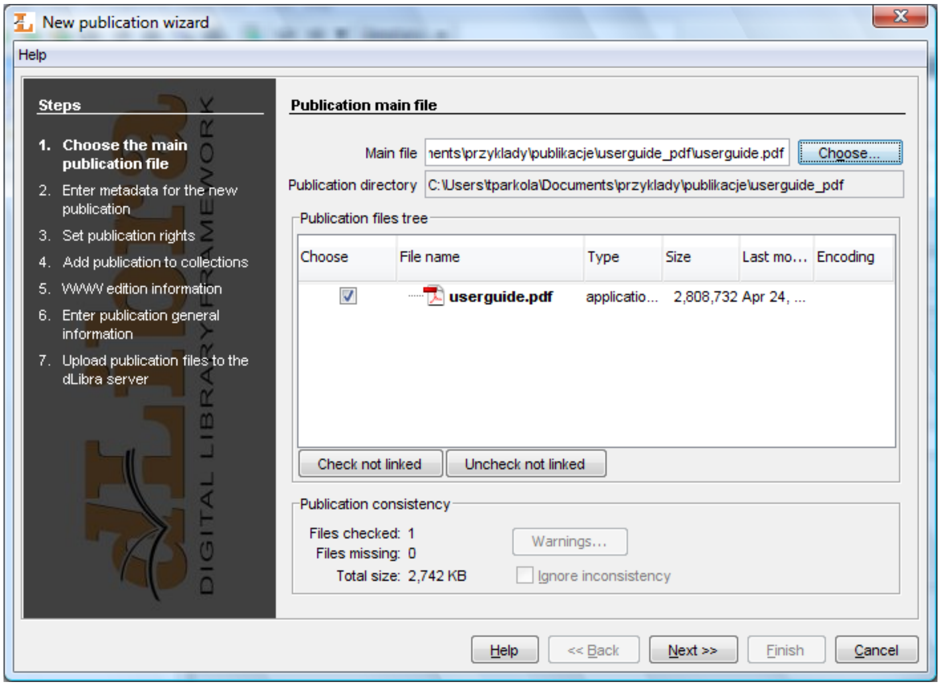
Beside publication files, all the other files in main file's directory are listed on the files table. By default these files are not checked to send to the server. However it is possible to add additional files.

Notice that the main publication file and cannot be unchecked and it will always be sent to the server.

On the publication files tree it is also possible to modify file's type and encoding. Type modification should be performed only if the application incorrectly identified it (provide correct value in Type column). Encoding modifications are possible only for files which have such information (e.g. HTML). If the application incorrectly identifies encoding the edtiro may correct it by selecting appropriate encoding from the list in Encoding column.

Press the Next button to move to the the page.

Figure 3.19. New Publication Wizard - main publication file page



Steps

1. **Choose the main publication file**
2. Enter metadata for the new publication
3. Set publication rights
4. Add publication to collections
5. WWW edition information
6. Enter publication general information
7. Upload publication files to the dLibra server

Publication main file

Main file: nents\przyklady\publikacje\userguide_pdf\userguide.pdf Choose...

Publication directory: C:\Users\tparkola\Documents\przyklady\publikacje\userguide_pdf

Publication files tree

Choose	File name	Type	Size	Last mo...	Encoding
☑	**userguide.pdf**	applicatio...	2,808,732	Apr 24, ...	

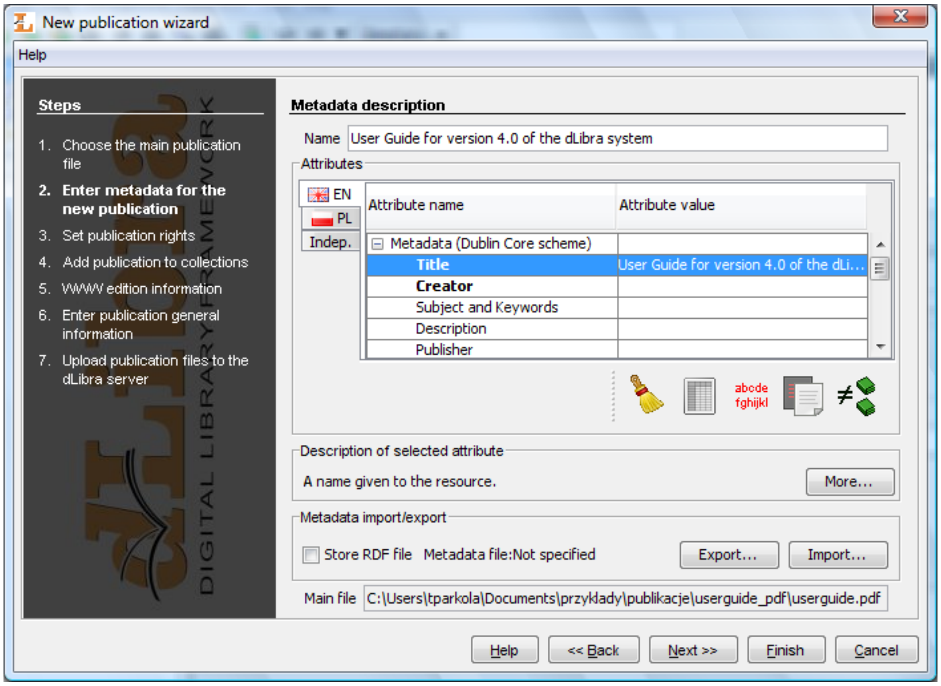
Publication consistency

Files checked: 1
Files missing: 0
Total size: 2,742 KB

□ Ignore inconsistency

4. In the second step (Figure 3.20, “New Publication Wizard - attribute values page”) the editor may provide metadata for the publication. See bibliographic description editor for details. In order to proceed to the next step publication name has to be given (Name field). By default the name is created basis on metadata by combining the title, author and publishing date (see configuration to change default settings). For example, if the title is “Stories”, author is “John Smith” and the publishing date is “2004” then the name will be “John Smith, 2004, Stories” - first author, then publishing date and title. Whenever user changes the metadata the name is updated. It is also possible to enter different name or modify default name. Press the Next button to move to the next step.

Figure 3.20. New Publication Wizard - attribute values page



Steps

1. Choose the main publication file
2. **Enter metadata for the new publication**
3. Set publication rights
4. Add publication to collections
5. WWW edition information
6. Enter publication general information
7. Upload publication files to the dLibra server

Metadata description

Name: User Guide for version 4.0 of the dLibra system

Attributes

Attribute name	Attribute value
⊟ Metadata (Dublin Core scheme)	
Title	User Guide for version 4.0 of the dLi...
Creator	
Subject and Keywords	
Description	
Publisher	

Description of selected attribute

A name given to the resource.

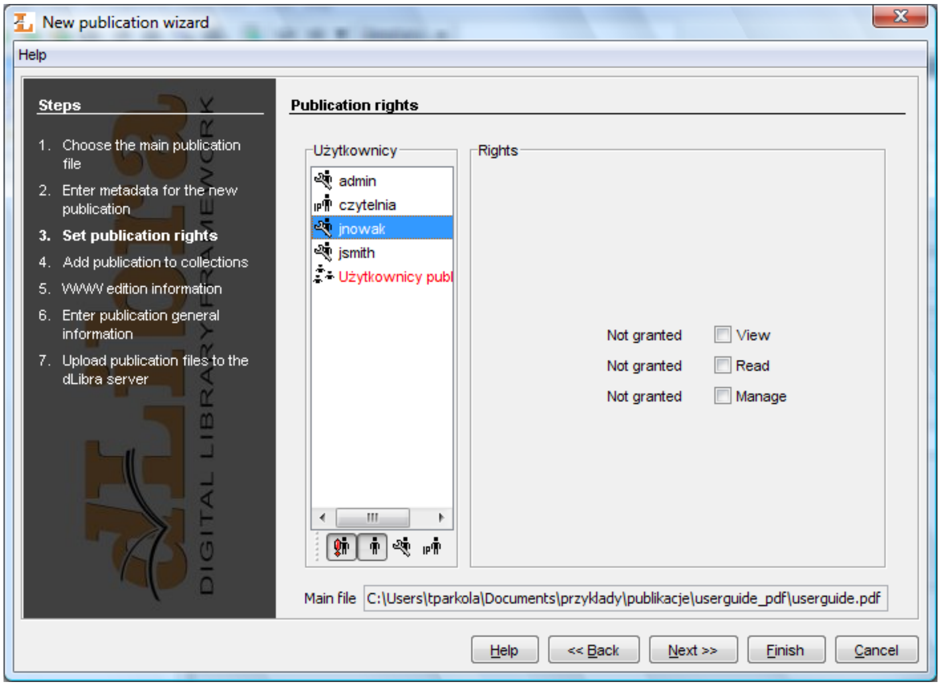
Metadata import/export

□ Store RDF file Metadata file:Not specified

Main file: C:\Users\tparkola\Documents\przykłady\publikacje\userguide_pdf\userguide.pdf

5. In the third step (Figure 3.21, “New Publication Wizard - publication rights page”) user may grant appropriate rights for other users to publication. For more details about rights management for publication see section Rights editor. Press the Next button to move to the next steps.

Figure 3.21. New Publication Wizard - publication rights page



Steps

1. Choose the main publication file
2. Enter metadata for the new publication
3. **Set publication rights**
4. Add publication to collections
5. WWW edition information
6. Enter publication general information
7. Upload publication files to the dLibra server

Publication rights

Użytkownicy
- admin
- czytelnia
- jnowak
- jsmith
- Użytkownicy publ

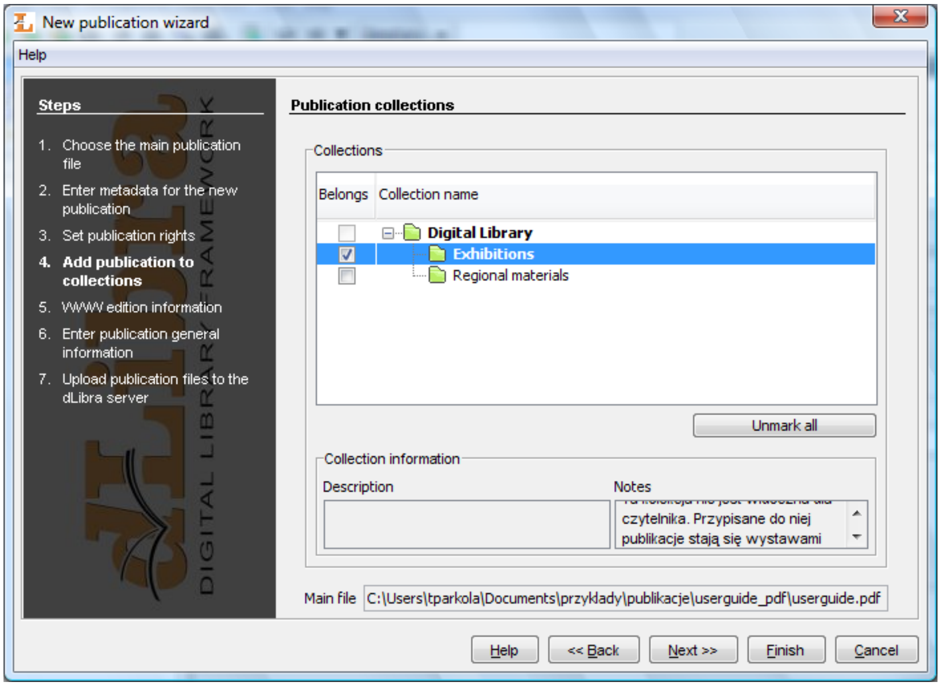
Rights

Not granted	□ View
Not granted	□ Read
Not granted	□ Manage

Main file: C:\Users\tparkola\Documents\przykłady\publikacje\userguide_pdf\userguide.pdf

6. In the fourth step (Figure 3.22, “New Publication Wizard - publication's collections page”) user may assign publication to collections. See Publication's collections editor for details. Press the Next button to move to the next step.

Figure 3.22. New Publication Wizard - publication's collections page



New publication wizard

Steps

1. Choose the main publication file
2. Enter metadata for the new publication
3. Set publication rights
4. **Add publication to collections**
5. WWW edition information
6. Enter publication general information
7. Upload publication files to the dLibra server

Publication collections

Collections

Belongs	Collection name
□	**Digital Library**
☑	Exhibitions
□	Regional materials

Collection information

Description:

Notes: czytelnika. Przypisane do niej publikacje stają się wystawami

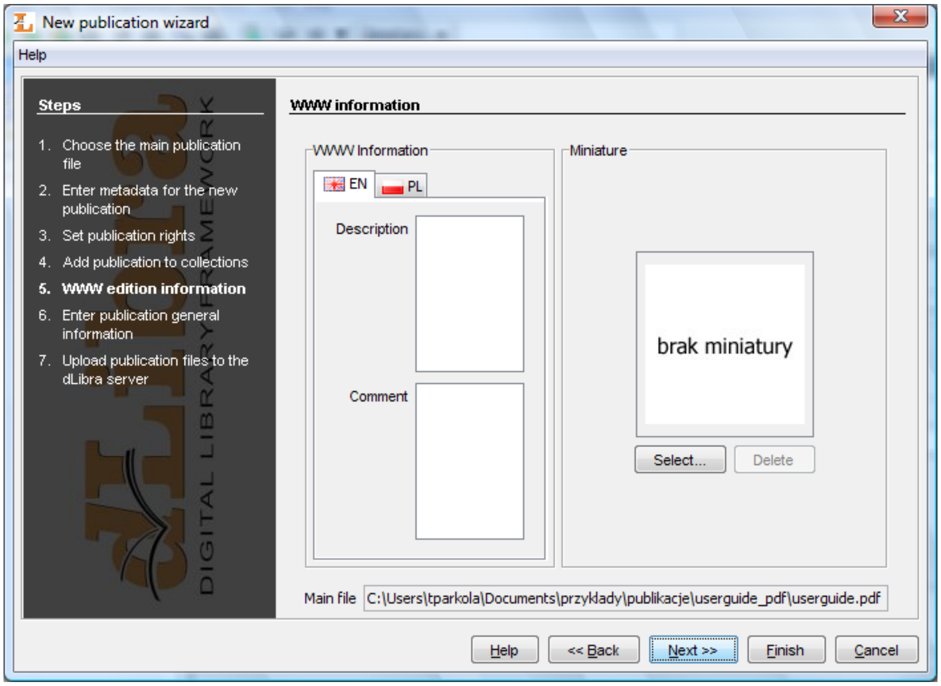
Main file: C:\Users\tparkola\Documents\przykłady\publikacje\userguide_pdf\userguide.pdf

7. In the fifth step (Figure 3.23, “New publication wizard - WWW information”) it is possible to specify WWW information concerning the first edition of the new publication. WWW information consists of:

   - Description - short edition description, e.g. what does it concern, visible on WWW search results page (it is recommended to provide the description possibly short --- one or two sentences),
   - Comment - characteristic information concerning the edition, e.g. specific marks (for example the lack of one page),
   - Miniature - graphic edition representation visible on the WWW edition information page.

   Press the Next button to move to the next step.

Figure 3.23. New publication wizard - WWW information



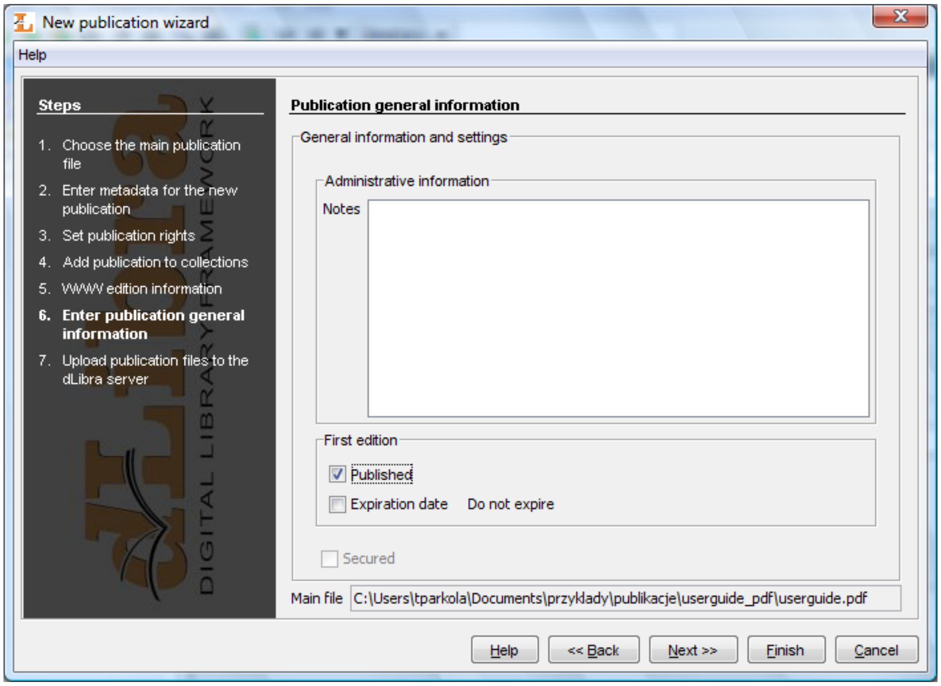
1. Choose the main publication file
2. Enter metadata for the new publication
3. Set publication rights
4. Add publication to collections
5. **WWW edition information**
6. Enter publication general information
7. Upload publication files to the dLibra server

Main file C:\Users\tparkola\Documents\przyklady\publikacje\userguide_pdf\userguide.pdf

8. Step sixth (Figure 3.24, “New publication wizard - general information”) allows user to provide general information:

   - Notes - administrative notes concerning publication
   - First edition - panel which allows for publishing the first edition (uncertain or time publishing)
   - Secured - if checked, publication will be secured - copying, printing will not be possible (applies only to HTML publications)

Figure 3.24. New publication wizard - general information



Steps

1. Choose the main publication file
2. Enter metadata for the new publication
3. Set publication rights
4. Add publication to collections
5. WWW edition information
6. **Enter publication general information**
7. Upload publication files to the dLibra server

Publication general information

General information and settings

Administrative information

Notes

First edition

☑ Published

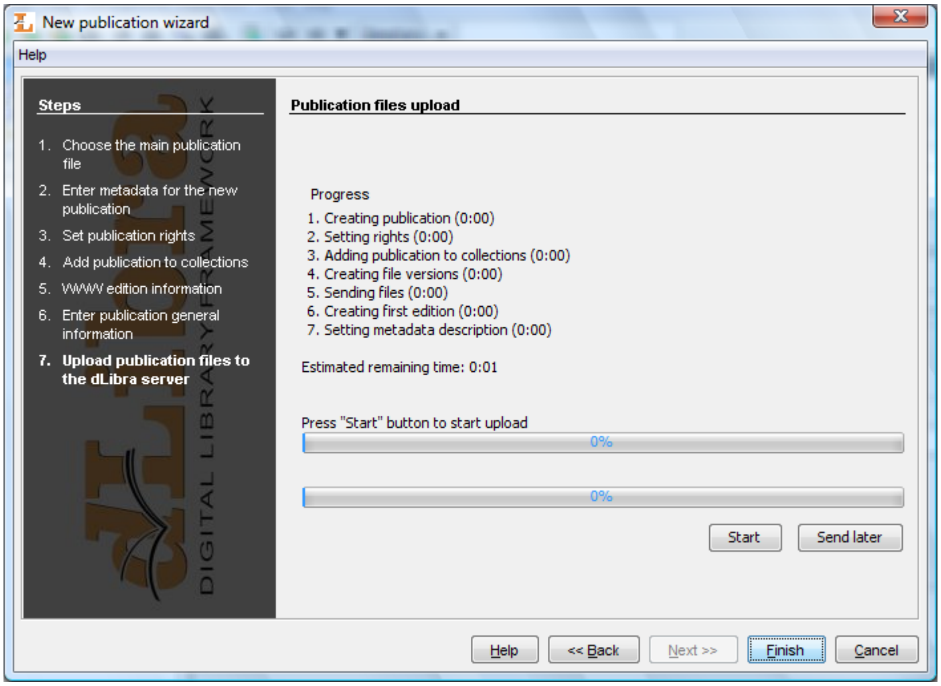
□ Expiration date Do not expire

□ Secured

Main file C:\Users\tparkola\Documents\przykłady\publikacje\userguide_pdf\userguide.pdf

9. Uploading files on the server is the last step (Figure 3.25, “New publication wizard - creating publication on the server”). Press the Start button to create publication on the server. The process can be paused and resumed by pressing the button again. During the creation process on the Progress panel all operations which have to be performed are presented. Operation which is being performed is marked with bold font. For every operation elapse time is measured and presented (in seconds) in the brackets after the operation name. Progress bars inform about progress of the operation. When all operations are finished a short summary is presented. During the creation process it is possible to cancel it (e.g. press the Cancel button). In addition, it is possible to postpone the whole creation process by pressing Send later button. When the Send later button is pressed all the information concerning publication are saved on the local drive and the wizard is immediatelly closed. All posponed publication are added to special list which can be managed on the send publication dialog which is described in the posponed publication upload mechanism section. Posponed publication upload mechanism allows editor to prepare a set of publications which are to be created and then start the upload of all the publications at once (e.g. start it at the end of the work). It should increase the number of submissions as the editor does not have to wait for each publication upload to complete because the upload of all prepared publications may proceed after editor's work.

Figure 3.25. New publication wizard - creating publication on the server



Steps

1. Choose the main publication file
2. Enter metadata for the new publication
3. Set publication rights
4. Add publication to collections
5. WWW edition information
6. Enter publication general information
7. **Upload publication files to the dLibra server**

Publication files upload

Progress

1. Creating publication (0:00)
2. Setting rights (0:00)
3. Adding publication to collections (0:00)
4. Creating file versions (0:00)
5. Sending files (0:00)
6. Creating first edition (0:00)
7. Setting metadata description (0:00)

Estimated remaining time: 0:01

Press "Start" button to start upload

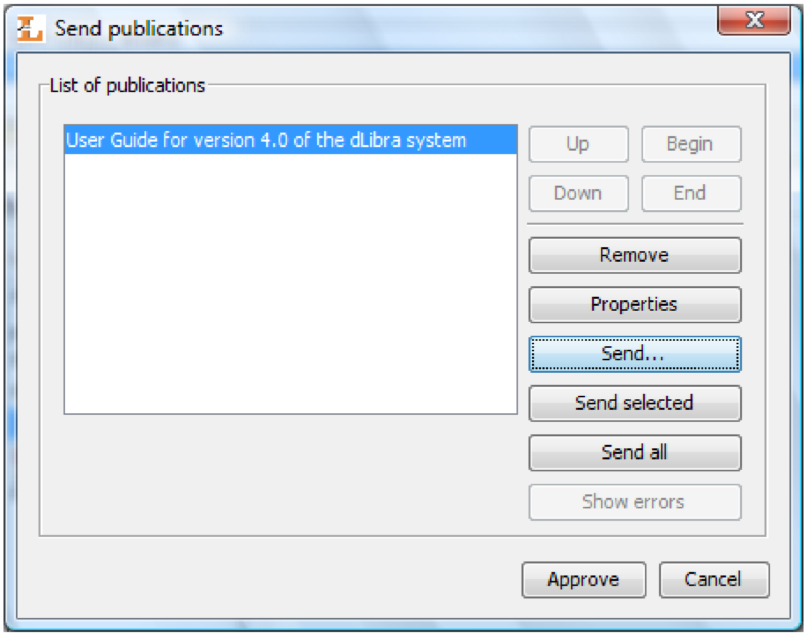
0%

0%

3.3.4. Sending posponed publications

In the last step of the new publication wizard the editor may choose Send later button. This option stores on the local drive all the information concerning prepared publication and closes the wizard window. This publication is not created in the system (the creation process is posponed) therefore the editor does not have to wait for the time-consuming upload process (especially when it comes to files which have large size files). The publication is added to special list of posponed publications. The list of posponed publications can be managed on the send publications dialog (Figure 3.26, “Send publications dialog”). In order to open this dialog select Send publications... option from Tools menu.

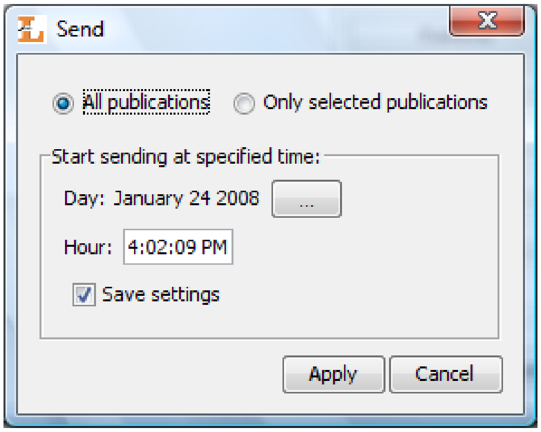
Figure 3.26. Send publications dialog



User Guide for version 4.0 of the dLibra system

The list of posponed publications is located on the left side of the send publications window. On the left side there is a set of buttons which allows editor to manage the list. To change the position of the publication selected on the list use Up, Down, Begin, End buttons. Delete button removes seleted publication from the list (all the information concerning the publication are irreversibly lost). Properties button allows editor to modify properties of selected publication. Send selected button starts the upload process of all publications selected on the list and opens the dialog where upload progress is displayed. The progress dialog may be minimized (upload progress is then displayed on the status bar - bottom right corner of the application window) which allows editor to proceed interaction with editor's application simultaneously to the running background publications upload process. In order to open the progress dialog again select Send publications... option from the Tools menu. Send all button starts the upload process of all the publications on the list. The progress dialog is opened. It can be minimized which allows editor to proceed interaction with editor's application simultaneously to the running background publications upload process. In order to open the progress dialog again select Send publications... option from the Tools menu. Send... button allows editor to specify the date and time in the future when the upload process is to be started. When the button is pressed the send parameters dialog (Figure 3.27, “Okno z parametrami wysyłania publikacji”) is opened where it is possible to provide additional send parameters.

Figure 3.27. Okno z parametrami wysyłania publikacji



Send

● All publications ○ Only selected publications

Start sending at specified time:

Day: January 24 2008

Hour: 4:02:09 PM

☑ Save settings

First the parameters specify whether all of the publications or only selected publications will be uploaded (respectively All publications and Only selected publications option). Second it is the date and time when the upload process is to be started. Third it is the Save settings check box which saves usef preferences so that when the next time this function is used start time (hour) will be the same and the start day will be distant from the present the same number of days. These parameters for exampe allow editors to balance the server load (e.g. each editor has concrete hour assigned when he/she should start the upload process of all prepared publication). When the parameters are applied the progress dialog is opened and it is displaying the time remaining to start the upload process. When remaining time elapses the upload process is started. This dialog may be minimized by the user which causes the countdown and the upload process to run in background. In order to open the progress dialog again select Send publications... option from the Tools menu.

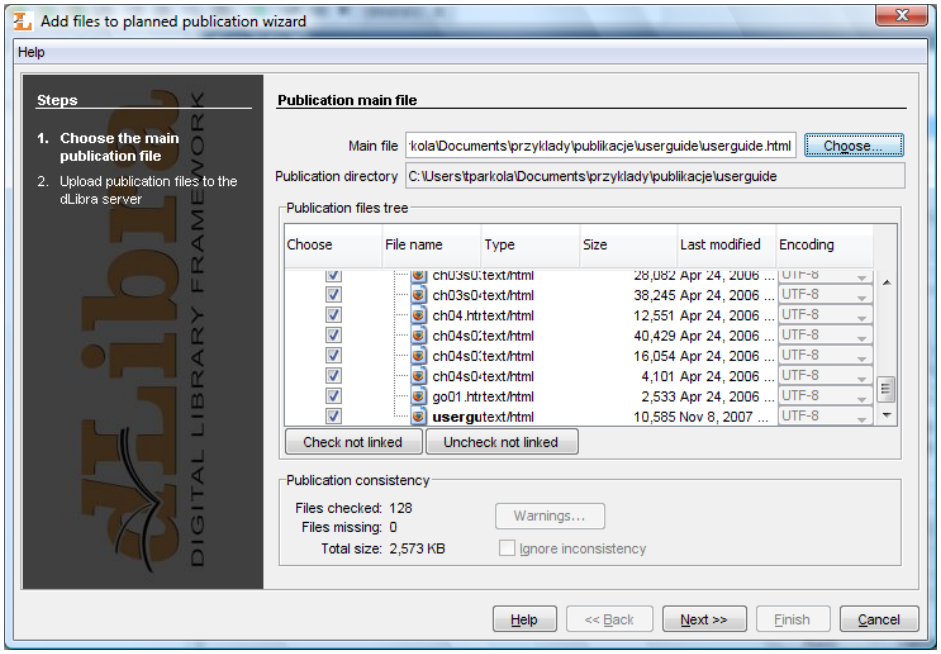
The last button (Show errors) is useful only if errors occurred during the upload process. Publication which could not be send are marked with error icon. To show errors connected with a publication, select it on the publications list and press Show errors button. The editor may try to correct publication properties (Properites button)in order to repair errors.

3.3.5. Creating a new planned publication

To create a new planned publication

1. On the elements list select destination directory or group publication
2. From the popup menu select New planned publication.... It is also possible to select this option from toolbar or main menu.
3. Planned publication wizard consist of four steps. The first step is the same as the second step of new publication wizard. The second step is the same as the third step of new publication wizard. The third step is the same as the fourth step of new publication wizard. In the last, fourth step of new planned publication wizard (Figure 3.28, “New planned publication wizard - general information”) specify the administrative notes and information whether the publication should be secured or not (Secured check box).

Figure 3.28. New planned publication wizard - general information


Add files to planned publication wizard
Help
Steps
1. Choose the main publication file
2. Upload publication files to the dLibra server
dLibra
DIGITAL LIBRARY FRAMEWORK
Publication main file
Choose...
Check not linked
Uncheck not linked
Warnings...
Help
<< Back
Next >>
Finish
Cancel

Main file ·kola\Documents\przyklady\publikacje\userguide\userguide.html

Publication directory C:\Users\tparkola\Documents\przyklady\publikacje\userguide

Publication files tree

Choose	File name	Type	Size	Last modified	Encoding
☑	ch03s0:	text/html	28,082	Apr 24, 2006 ...	UTF-8
☑	ch03s0·	text/html	38,245	Apr 24, 2006 ...	UTF-8
☑	ch04.htr	text/html	12,551	Apr 24, 2006 ...	UTF-8
☑	ch04s0:	text/html	40,429	Apr 24, 2006 ...	UTF-8
☑	ch04s0:	text/html	16,054	Apr 24, 2006 ...	UTF-8
☑	ch04s0·	text/html	4,101	Apr 24, 2006 ...	UTF-8
☑	go01.htr	text/html	2,533	Apr 24, 2006 ...	UTF-8
☑	**usergu**	text/html	10,585	Nov 8, 2007 ...	UTF-8

Publication consistency

Files checked: 128
Files missing: 0
Total size: 2,573 KB

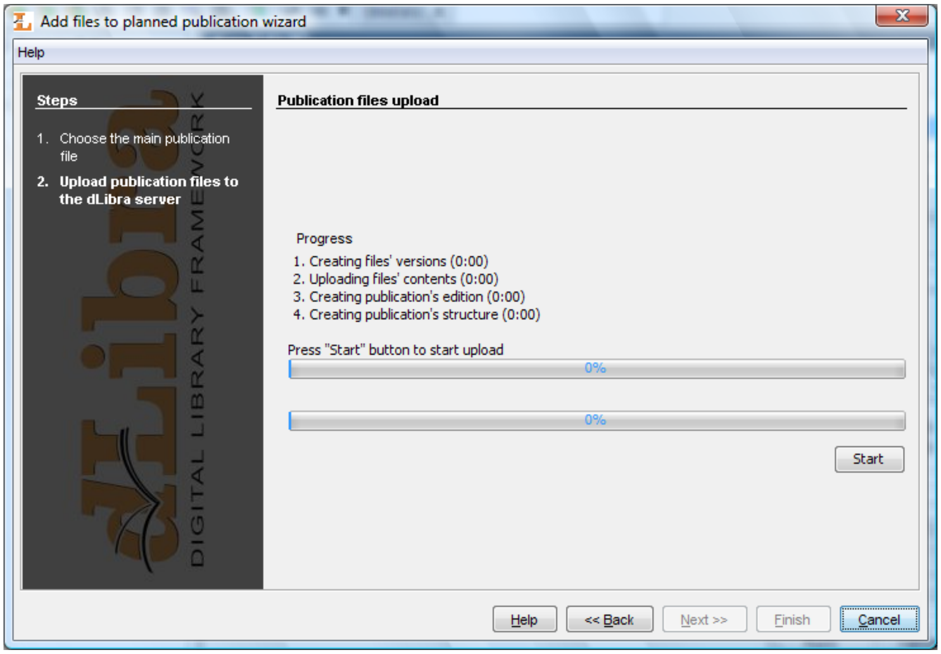
□ Ignore inconsistency

3.3.6. Adding content to planned publication

To add files to planned publication:

1. On the elements list select a planned publication which you want to add files to.
2. From the popup menu select Add files....
3. Add content wizard consists of two stepts. The first step is the same as the firts step of new publication wizard. The second step (Figure 3.29, “Add content wizard - sending content (files) on the server”) allows sending publication content on the server. In order to send selected files in the first step press Start button.

Figure 3.29. Add content wizard - sending content (files) on the server



Add files to planned publication wizard

Steps

1. Choose the main publication file
2. **Upload publication files to the dLibra server**

Publication files upload

Progress

1. Creating files' versions (0:00)
2. Uploading files' contents (0:00)
3. Creating publication's edition (0:00)
4. Creating publication's structure (0:00)

Press "Start" button to start upload

3.3.7. Publishing publication's edition

In order that reader could read publication's edition, it is required to assign to publication a “read” right for him and publish the edition. In order to publish the edition and assign a “read” right to the reader:

1. On the elements list select a publication which edition is to be visible for the reader.
2. On the edition's general information tab (Edition) select Published option.
3. On the rights assignment tab (Rights) select the user name (or group name - if we want to assign right to the group of users) which you want to grant the “read” right and then select the View check box.

In order to approve the changes press the Apply button. The publication is now visible for users which have “read” right assigned.

3.3.8. Moving and removing elements

To remove an element:

1. In the directories tree or on the elements list select an element to remove.
2. From the popup menu select 🗑 Delete option and confirm operation. Alternatively, a main menu or toolbar (only in case of an element selected on elements list) can be used.

   In case of publication and directory which contains at least one publication it is possible to specify remove reason for publications. The remove reason is presented to the WWW if a user attempts to read publication.

There are two ways of moving an object. The first way is possible when editor application has two elements list visible (see???). In order to move selected element from one list to the open element of the second list press **F6** key. Remember that moving huge amounts of object (eg. directory) may cause high server load caused by information actualization. The second way is to use “Drag and drop” (this way you can move element between elements list and directories tree).

3.3.9. Deleting publication's content

Content deleting is useful when we want to remove publication content (files), but leave editions' bibliographic description and delete reason.

To delete publication's content:

1. On the elements list select a publication which content is to be deleted,
2. From the popup menu select Delete content... option.
3. Enter delete reason. Attention! The reason is required because it is presented when WWW reader visits publication page.

3.3.10. Changing password

Every user of the editor and administrator program may change his/her password. In order to change password press Change password... option from Program menu. Change password window will appear. To change password provide current password, new password and confirm new password.

3.3.11. Searching

User may find an element in two different ways.

- Searching for an element basis on bibliographic description or administrative notes.
- Searching for an element basis on its identifier.

Searching basis on bibliographic description or administrative notes gives two options. First it is searching in bibliografic description which gives the same results as searching on *dLibra* web pages. Second it is searching in administrative notes of elements. In order to open search window where both options are available select Find... option from Edit menu.

To run searching basis on identifier select Go to... option from Edit menu. Then select type of an element to find (e.g. directory) and type identifier of the element to find. If such an element exists it will be displayed and selected on elements list. This kind of search is also available on the tool bar in editor's application where it is possblie to choose element's type and provide an identifier (in order to start searching press ENTER on the keyboard).

3.4. Advanced editor's tasks

This section describes advanced tasks performed by the editor.

3.4.1. Editor's application modes (views)

In the editor's application two work modes are possible:

- Simple mode
- Advanced mode

By default the simple mode is enabled. In order to switch to the advanced mode select the Advanced mode option from the View menu. In ordert to get back to the simple mode select again the Adcanced mode option from the View menu.

The differences between the modes (views) are presented on the Table 3.1, “Differences between editor's application work modes”.

Table 3.1. Differences between editor's application work modes

Feature \ Mode	Simple mode	Advanced mode
Publication node	• If the publication has only one edtion then after opening it the edition is not visible - only files versions of the edition are visible. All properties of the edition and publication can be viewed/modified on the tabs in the properties window after the publication is selected on the elements list, • If the publication has more that one edition - all editions and publication files element are visible after opening the publication (it is the same situation as in the advanced mode). If the publication is selected on the elements list then on the properties window tabs only publication properties is available. In order to view/modify properties of particular edition, open the publication and select the edition on the elements list (edition properties will be presented on the properties window).	Afrter opening a publication all its editions are visible. Files versions which compose particular edition are visible after opening the edition. On the elements list there is also a publication files element which contains all files connected with the publication.

3.4.2. Downloading an edition of a publication

In order to further edit an existing publication, its files must be downloaded into editor's local machine.

To download an edition of a publication:

1. On the elements list select the edition to be downloaded (it can be the current edition as well).

2. From the context menu select the Download... function. Alternatively, the toolbar button or main menu options can be used.

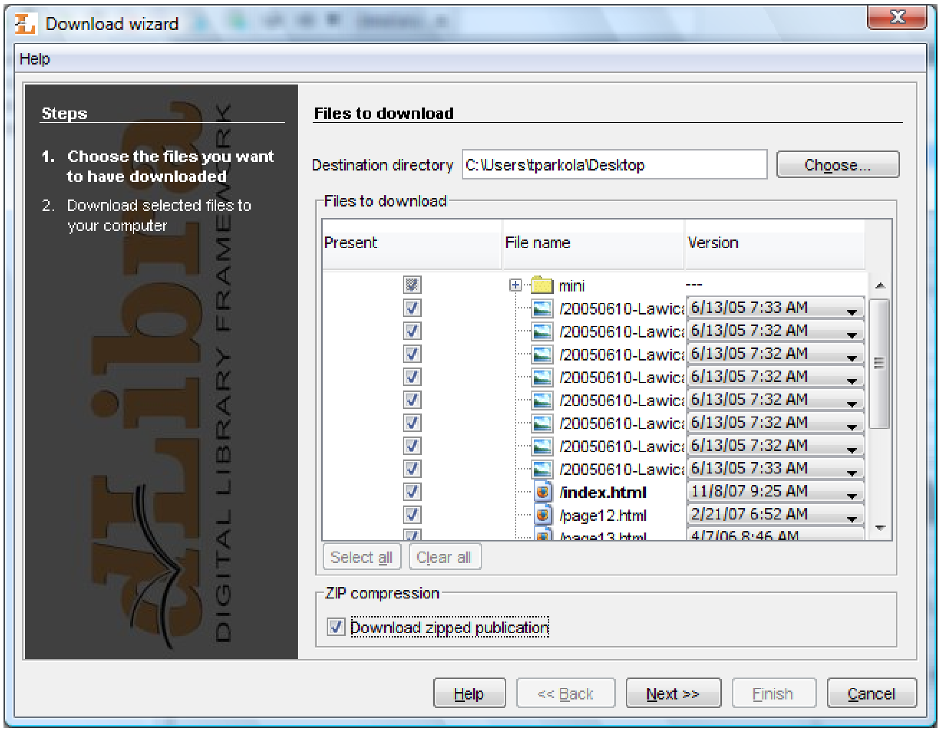
3. In the first step (Figure 3.30, “Download Wizard - publication file selection page”) select the local disk directory the publication files shall be stored in.

   In the Files to download table the default file selection can be altered. To exclude a file from the download list --- uncheck the appropriate position in the Present column. Additionally, the version to be downloaded can be changed using a pull down list in the Version column.

   It is possible to download the publication as the zip archive. In order to do this check Download zipped publication check box.

   Press the Next button to proceed to the file download page.

Figure 3.30. Download Wizard - publication file selection page



Destination directory: C:\Users\tparkola\Desktop

Files to download

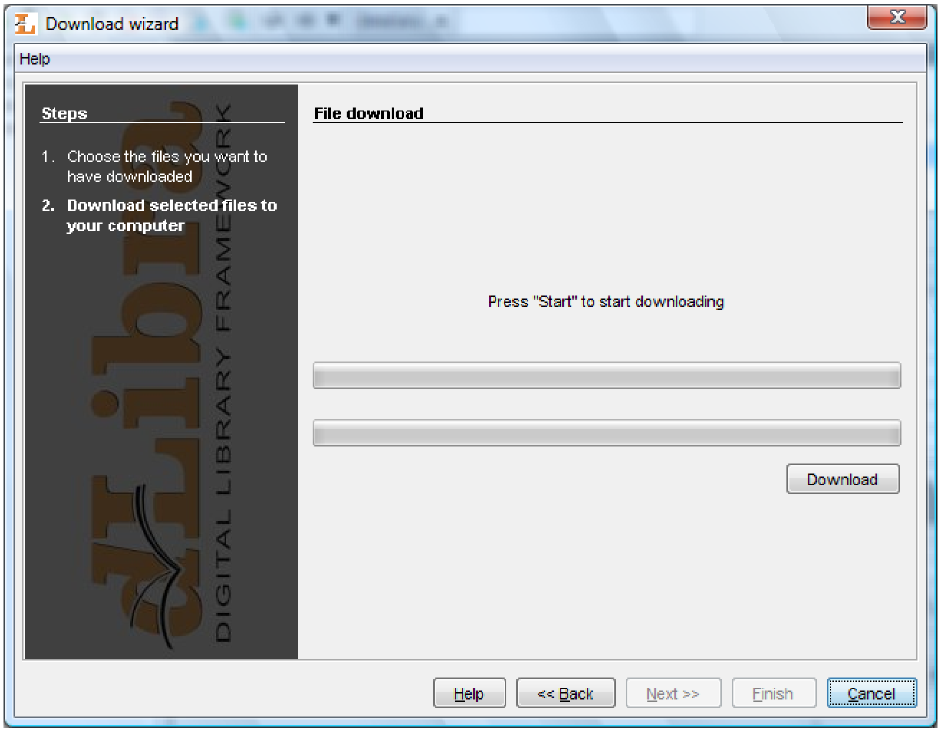
Present	File name	Version
☑	mini	---
☑	/20050610-Lawic:	6/13/05 7:33 AM
☑	/20050610-Lawic:	6/13/05 7:32 AM
☑	/20050610-Lawic:	6/13/05 7:32 AM
☑	/20050610-Lawic:	6/13/05 7:32 AM
☑	/20050610-Lawic:	6/13/05 7:32 AM
☑	/20050610-Lawic:	6/13/05 7:32 AM
☑	/20050610-Lawic:	6/13/05 7:32 AM
☑	/20050610-Lawic:	6/13/05 7:33 AM
☑	**/index.html**	11/8/07 9:25 AM
☑	/page12.html	2/21/07 6:52 AM
☑	/page13.html	4/7/06 8:46 AM

☑ Download zipped publication

4. In the second step (Figure 3.31, “Download wizard - file download page”) press the Start button to start downloading the selected publication files to the selected local disk directory. The process can be paused and resumed pressing the button again.

   After the downloading process finishes press the Finish button to close the wizard.

Figure 3.31. Download wizard - file download page



Steps

1. Choose the files you want to have downloaded
2. **Download selected files to your computer**

File download

Press "Start" to start downloading

3.4.3. Adding new files' versions

If the author of already published publication will modify/update its content then it is needed to update this publication in the dLibra system (modified files have to be sent to the server). After the update operation it is possible to make those changes available for web users (new edition can be created based on newly added filed).

To send the new files to the server:

1. Switch the application to the advanced mode then on the elements list open the publication to which the new files versions shall be assigned and select *Publication files* element.

2. From the context menu select the Add new files' versions... function.

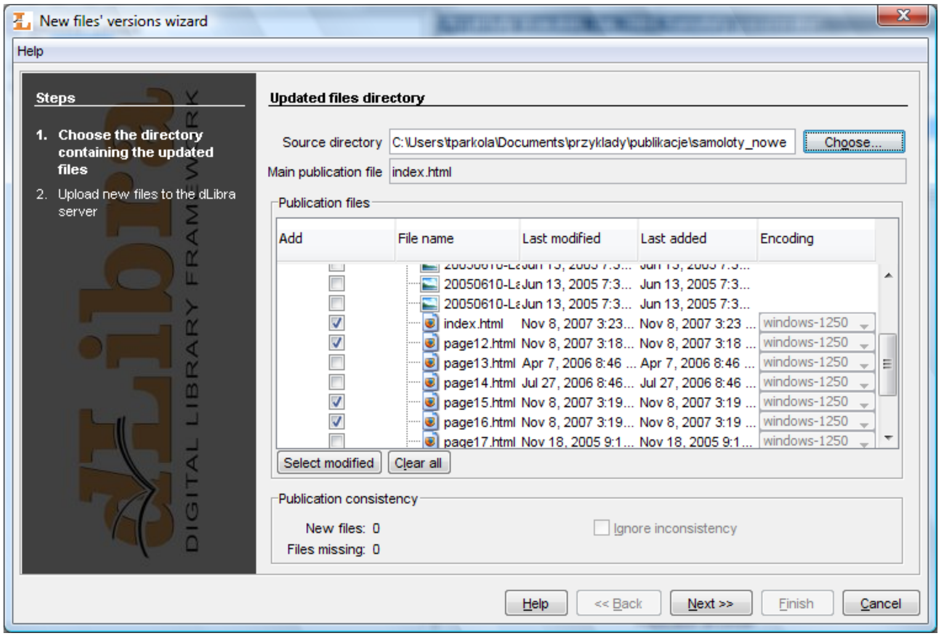
3. In the Add new files' versions wizard select the local disk directory that contains the updated publication files.

   Selected directory has to contain publication main file. Missing files are marked by .

   The wizard selects files to update based on the modification and storage dates. By default, files modified after the last store/add operation are selected. The selection can be altered by checking/unchecking appropriate positions in the Add column of the publication files table.

   Press the Next button to proceed to the file upload page.

Figure 3.32. File Update Wizard - updated files directory page



1. Choose the directory containing the updated files
2. Upload new files to the dLibra server

Source directory: C:\Users\tparkola\Documents\przyklady\publikacje\samoloty_nowe

Main publication file: index.html

Add	File name	Last modified	Last added	Encoding
□	20050610-La	Jun 13, 2005 7:3...	Jun 13, 2005 7:3...	
□	20050610-La	Jun 13, 2005 7:3...	Jun 13, 2005 7:3...	
□	20050610-La	Jun 13, 2005 7:3...	Jun 13, 2005 7:3...	
☑	index.html	Nov 8, 2007 3:23...	Nov 8, 2007 3:23 ...	windows-1250
☑	page12.html	Nov 8, 2007 3:18...	Nov 8, 2007 3:18 ...	windows-1250
□	page13.html	Apr 7, 2006 8:46 ...	Apr 7, 2006 8:46 ...	windows-1250
□	page14.html	Jul 27, 2006 8:46...	Jul 27, 2006 8:46 ...	windows-1250
☑	page15.html	Nov 8, 2007 3:19...	Nov 8, 2007 3:19 ...	windows-1250
☑	page16.html	Nov 8, 2007 3:19...	Nov 8, 2007 3:19 ...	windows-1250
□	page17.html	Nov 18, 2005 9:1...	Nov 18, 2005 9:1...	windows-1250

New files: 0

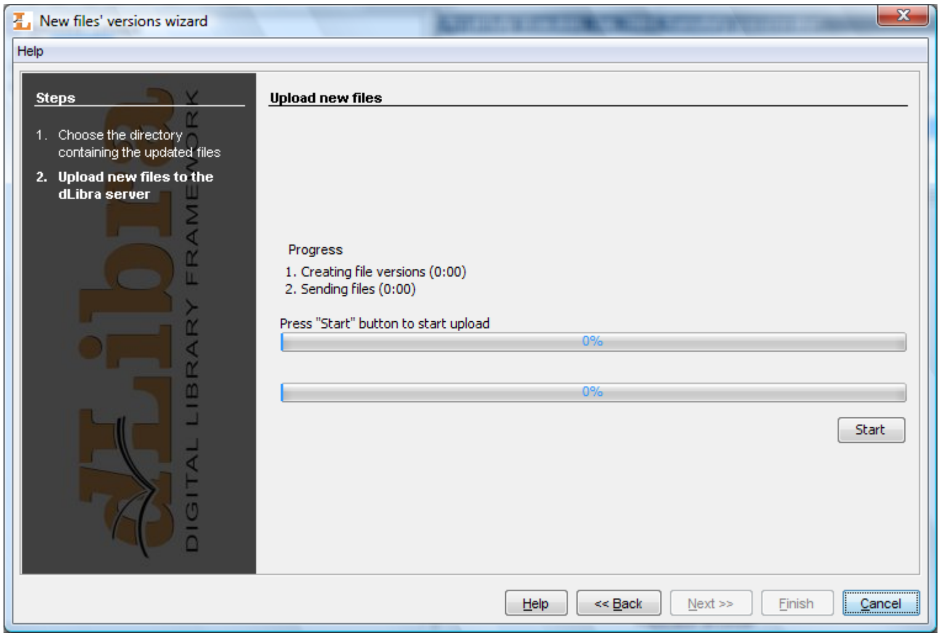
Files missing: 0

□ Ignore inconsistency

4. Press the Start button to start uploading selected publication files to the server. The process can be paused and resumed by pressing the button again.

   After the uploading process finishes press the Finish button to close the wizard.

Figure 3.33. File Update Wizard - file upload page



1. Choose the directory containing the updated files
2. Upload new files to the dLibra server

Progress

1. Creating file versions (0:00)
2. Sending files (0:00)

Press "Start" button to start upload

3.4.4. Creating/removing an edition of a publication

Edition is a viewable form of a publication - to make a publication available for viewing an edition must be created and published.

To create an edition of a publication:

1. On the elements list select the edition (e.g. the current edition) the new edition should be based on.

2. From the context menu select the New edition... function. Alternatively, the toolbar button or main menu options can be used.

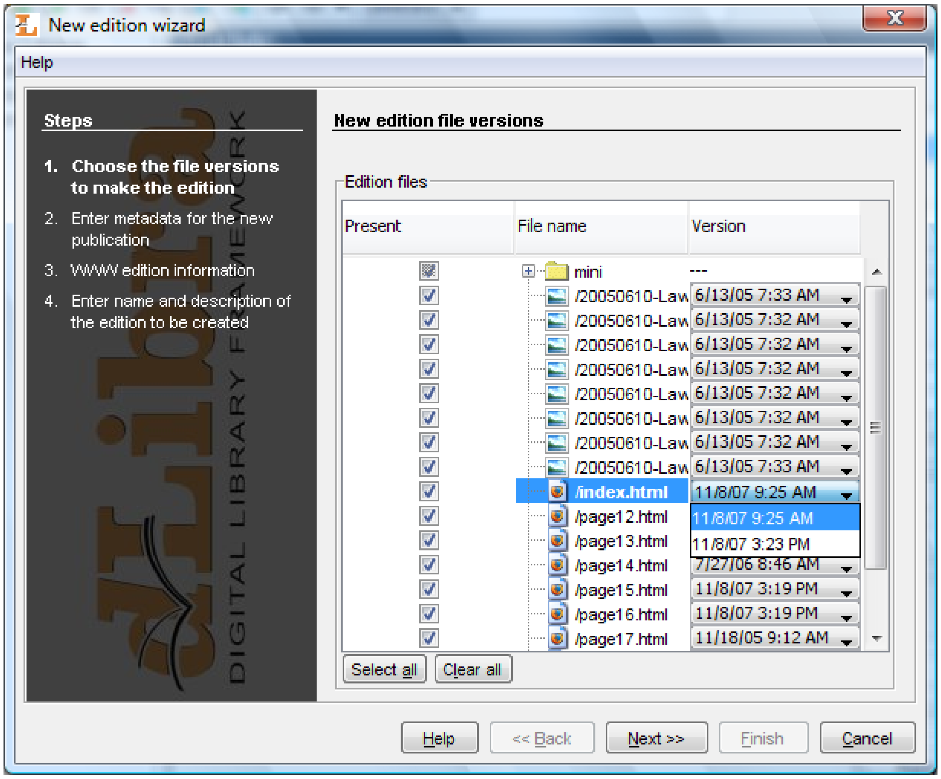
3. In the New Edition Wizard select the versions of publication files to compose the new edition.

   When the new edition is based on a branch of a publication, the newest versions of files lying on the branch are selected. When the new edition is based on an existing edition of a publication the selected file versions are the same as in the base edition.

   The selection of file versions can be altered using a drop down list embedded in the Version column of the Edition files table.

   To proceed to the edition properties page press the Next button.

   Figure 3.34. New Edition Wizard - file versions selection page



Steps

1. Choose the file versions to make the edition
2. Enter metadata for the new publication
3. WWW edition information
4. Enter name and description of the edition to be created

New edition file versions

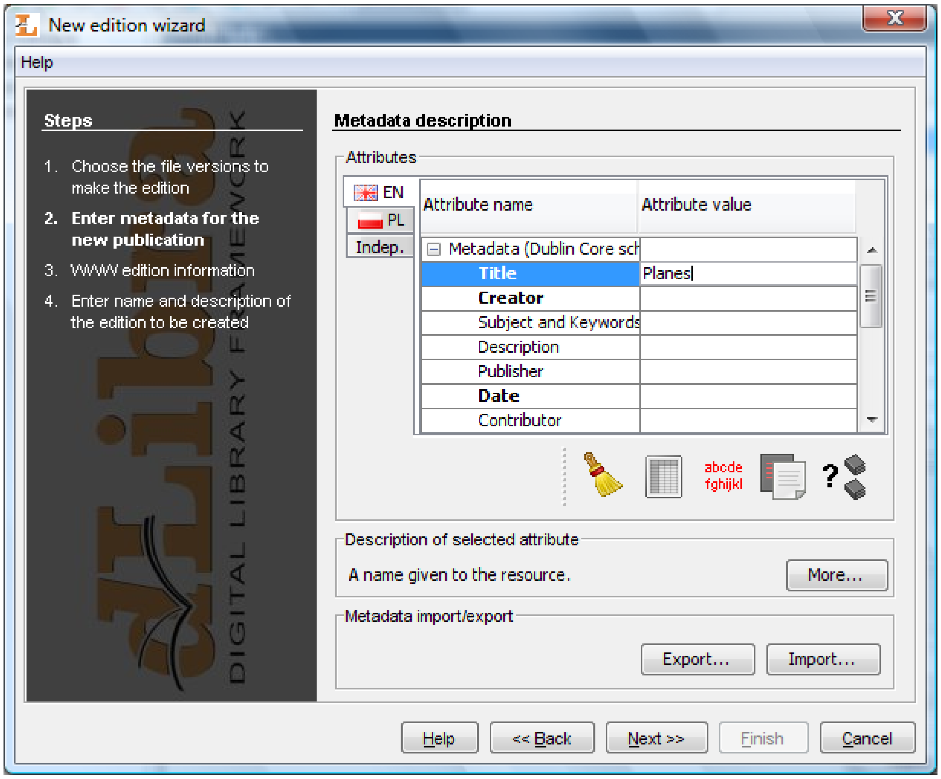
Edition files

Present	File name	Version
▨	mini	---
☑	/20050610-Law	6/13/05 7:33 AM
☑	/20050610-Law	6/13/05 7:32 AM
☑	/20050610-Law	6/13/05 7:32 AM
☑	/20050610-Law	6/13/05 7:32 AM
☑	/20050610-Law	6/13/05 7:32 AM
☑	/20050610-Law	6/13/05 7:32 AM
☑	/20050610-Law	6/13/05 7:32 AM
☑	/20050610-Law	6/13/05 7:33 AM
☑	/index.html	11/8/07 9:25 AM
☑	/page12.html	11/8/07 9:25 AM
☑	/page13.html	11/8/07 3:23 PM
☑	/page14.html	7/27/06 8:46 AM
☑	/page15.html	11/8/07 3:19 PM
☑	/page16.html	11/8/07 3:19 PM
☑	/page17.html	11/18/05 9:12 AM

4. In this step enter bibliographic description for new edition. It is possible to import bibliographic description from *MARC 21* lub *RDF* format.

To proceed to the next step press Next button.

Figure 3.35. New Edition Wizard - bibliographic description page



Steps

1. Choose the file versions to make the edition
2. **Enter metadata for the new publication**
3. WWW edition information
4. Enter name and description of the edition to be created

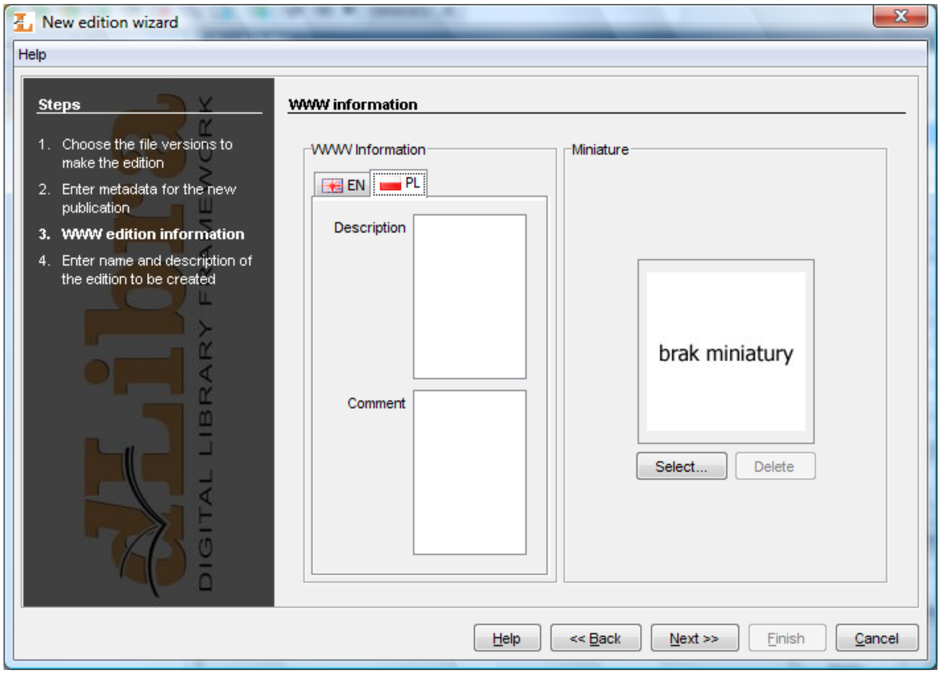
Attribute name	Attribute value
⊟ Metadata (Dublin Core sch	
Title	Planes|
Creator	
Subject and Keywords	
Description	
Publisher	
Date	
Contributor	

A name given to the resource.

5. Enter WWW information: Description which is presented on search results list, Comment available on edition information page and miniature presented on edition information page.

   To proceed to the next step press Next button.

Figure 3.36. New Edition Wizard - WWW information



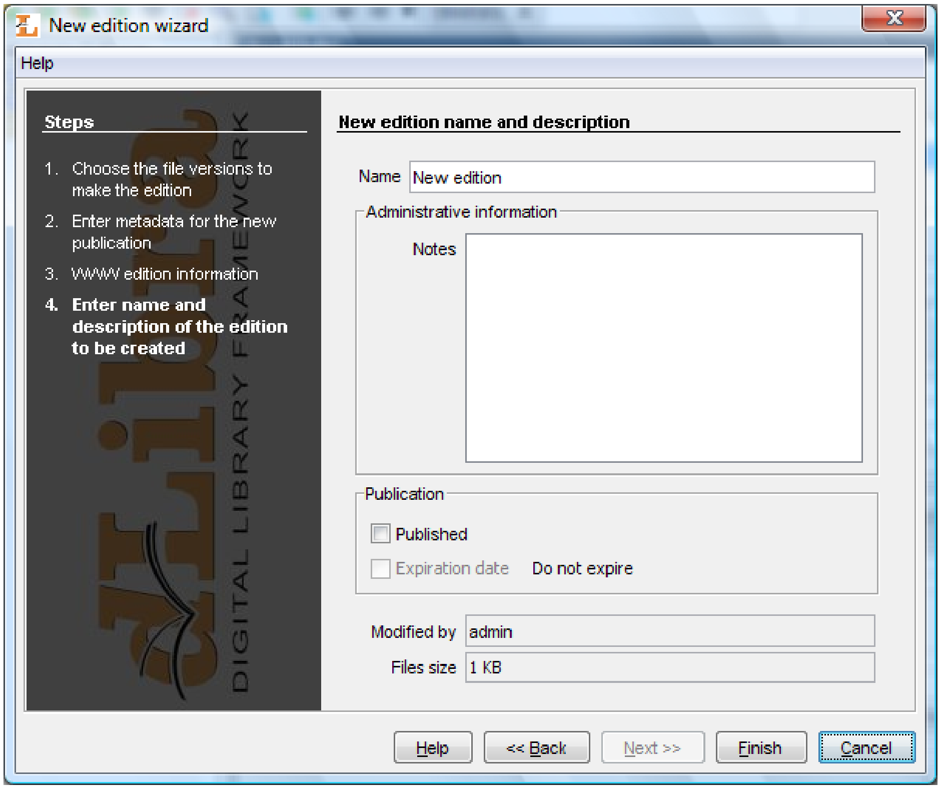
Steps

1. Choose the file versions to make the edition
2. Enter metadata for the new publication
3. **WWW edition information**
4. Enter name and description of the edition to be created

6. Enter the new edition's name and description in the Name and Description fields. To make the new edition published check the Published option. This can also be done after the edition is created.

   To create the new edition press the Finish button.

Figure 3.37. New Edition Wizard - edition properties page



1. Choose the file versions to make the edition
2. Enter metadata for the new publication
3. WWW edition information
4. **Enter name and description of the edition to be created**

New edition name and description

Name: New edition

Administrative information

Notes:

Publication

□ Published

□ Expiration date Do not expire

Modified by: admin

Files size: 1 KB

In order to remove edition two condition have to be fulfilled:

- Edition is not the only one edition of publication
- Edition is not published

Remove edition choosing Delete option from context menu.

3.4.5. Managing publication files

There are three mechanisms which allow publication files management. The first one is file version removal. File version can be removed when it is not the only one version of a file and it is does not belong to any edition. The second mechanism is file removal. File can be removed when none of its versions belongs to any edition. The third mechanism is automatic deletion of redundant files. Redundant files of publication are files and files versions which may be removed. In order to delete redundant files choose Remove redundant files... option from context menu of Publication files element (it is available after publication is opened, only in advanced mode).

3.4.6. Mass publication upload

In this section it is described mass publication upload mechanism.

3.4.6.1. Mass publication upload wizard

To start mass publication upload creator select Mass publication upload from Tools menu.

1. In the first step (Figure 3.38, “Mass publication upload creator - publications selection.”) we specify publications which will be sent to *dLibra* server.

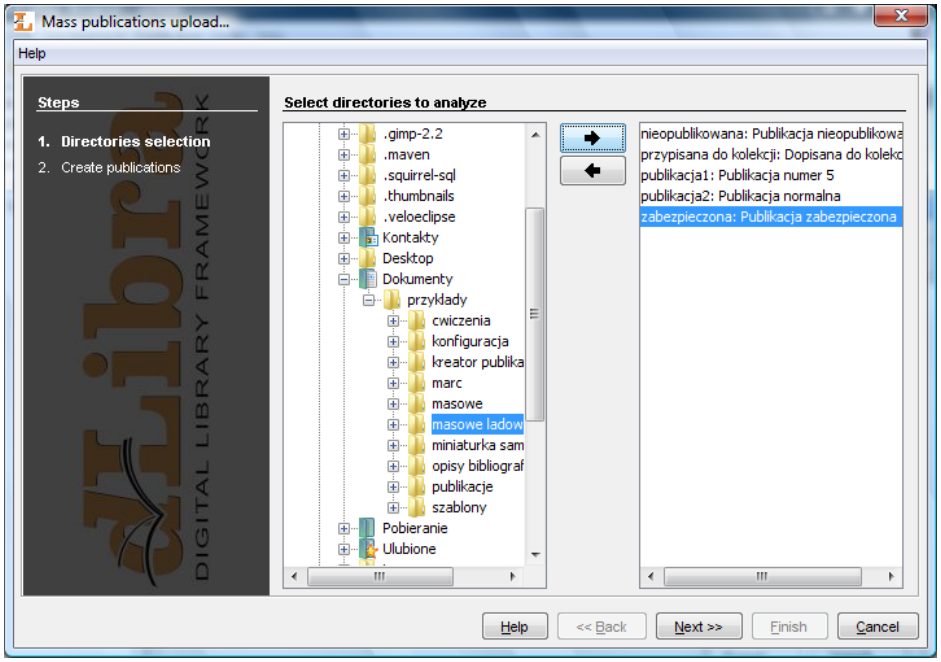
   First step's window is splitted into two parts: on the left side there is a system directories tree, on the right side there is a list of publication to upload. In order to add publication to upload:

   a. On the system directories tree check a directory containing subdirectories describing publications (information about the structure of a directory describing a publication are placed in the next section).

   b. Press the ➡ button. Application will check corectness of directories describing publications. In case of errors, appropriate messages will be shown.

   After adding publications to send it is possible to go to the next step.

   Figure 3.38. Mass publication upload creator - publications selection.

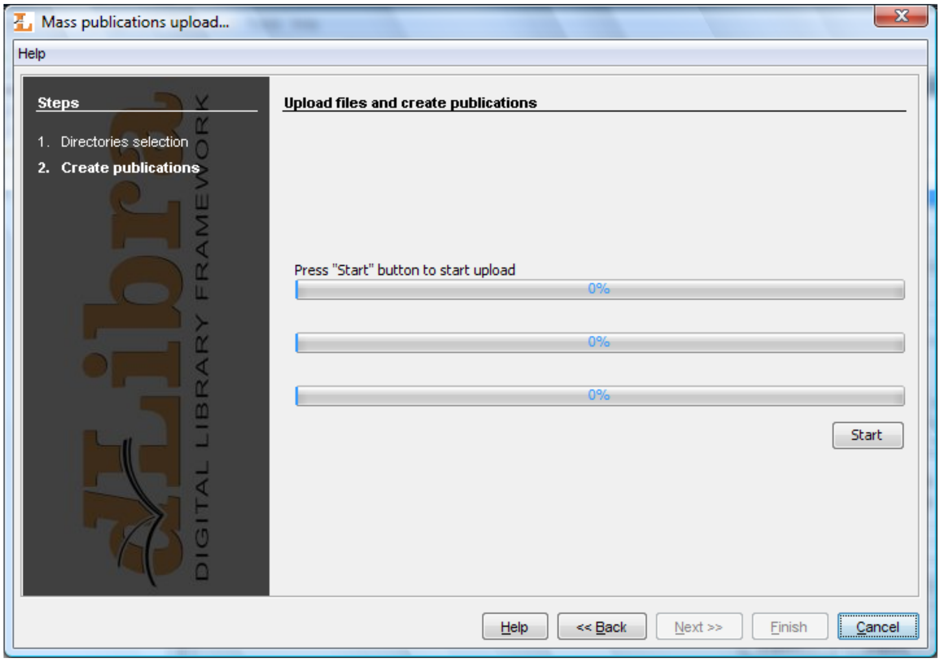
   

   nieopublikowana: Publikacja nieopublikowa
   przypisana do kolekcji: Dopisana do kolekc
   publikacja1: Publikacja numer 5
   publikacja2: Publikacja normalna
   zabezpieczona: Publikacja zabezpieczona

2. In the second step we send publication to the server.

   Figure 3.39, “Mass publication upload creator - publications upload” shows a window of the second step. To send publications press Start button. During the upload process is it is possible to monitor the progress throught to the progress bars. When the process finished appropriate summary is shown. If for any reason uploading of one or more publications failed, it is possible to check error messages by pressing the Errors details... button.

Figure 3.39. Mass publication upload creator - publications upload


Mass publications upload...
Help
Steps
1. Directories selection
2. Create publications
dLibra DIGITAL LIBRARY FRAMEWORK
Upload files and create publications
Press "Start" button to start upload
0%
0%
0%
Start
Help
<< Back
Next >>
Finish
Cancel

3.4.6.2. Structure of a directory describing a publication

Directory describing a publication consists of:

- `publication.properties` file which describes publication,
- Optional RDF file which contains bibliographic description fo the publication,
- Publication files.

3.4.6.2.1. `publication.properties` file

`publication.properties` file is publication properties file. It consists of attribute—value pairs. Attribute is separated from value by the equals sign (=). Every pair is placed in separate line.

Possible attributes and their values are presented below.

- Attribute publication.actorsRights.<user login or group name to which we will assign rights>

  Rights separated with a comma. The list of possible rights:

  - pv – publication view
  - pr – publication read
  - pe – publication management (edit)

- Attribute publication.collections

Collections identifiers separated by a comma to which publication is be assigned to.

- Attribute publication.destination.directoryId

  Destination directory identifier.

- Attribute publication.destination.parentPublicationId

  Destination group publication identifier.

- Attribute publication.mainFile

  The name of the publication main file (including extension). Specified file has to be in directory describing this publication.

- Attribute publication.metadataFile

  Metadata filename (including extension). Metadata file has to be accordant with RDF format and it has to be placed in directory describing a publication. By default metadata filename is the same as publication main file name except the extension—it is `.rdf`.

- Attribute publication.name

  Publication name.

- Attribute publication.notes

  Administrative notes concerning publication.

- Attribute publication.published

  Attribute indicating whether the firs edition of this publication should be published or not. If the value is true edition will be published. In any other case the edition will not be published. Default value is false.

- Attribute publication.secured

  Attribute indicating whether the publication should be secured or not. If the value is true publication will be secured. In any other case it will not be secured. Default value is false.

- Attribute edition.image.content

  The name of the file which contains edition's miniature.

Examplary `publication.properties` file:

```
publication.actorsRights.publicGroup=pv,pr
publication.collections=2,7,12
publication.destination.directoryId=1
publication.mainFile=index.htm
publication.metadataFile=metadata.rdf
publication.name=Przykładowa publikacja
publication.notes=Stworzone przez Jana Kowalskiego
publication.published=true
```

Above examplary file specifies a publication. *publicGroup* will have right to view (pv) and read (pr) it. Publication will be assigned to collections indicated by identifier 2, 7 and 12. Publication will be placed in the directory indicated by identifier 1. Publication main file will be `index.htm` and metadata are in `metadata.rdf` file. Publication name is “Examplary publication” and administrative notes are “Created by John Smith”. First edition of this publication will be published and publication itself will not be secured.

To create publication it is required a minimal set of informations described below. It it is impossible to gain all these informations it is impossible to creata a publication.

- Publication destination in Directories Tree. There are two ways to specify it: destination directory (publication.destination.directoryId attribute) and destination group publication (publication.destination.parentPublicationId attribute) or destination directory only.
- Publication main file. It is specified by publication.mainFile attribute and it is required.
- Publication name. There are two ways to specify publication name. First way is to specify publication.name attribute. If this attribute is not specified metadata file will be checked in order to compose publication name in the same way as in fourth step of new publication creator (it is composed of title, author and date). So metadata file is the second way.

3.4.6.2.2. Inheritance in `publication.properties` files

Inheritance mechanism was introduced in order to simplify assigning some attributes from `publication.properties` file.

Inheritance mechanism works in the following way: For every `publication.properties` file, which is to be read (let us name it a “child”), it is checked if there exists a `publication.properties` file in the upper directory (let us name it a “parent”). If the “parent” file exists its content is read, then the content of “child” is read. During the reading process there is a rule that says that attribute values are overridden. It means that if both “parent” file and “child” file specify the same attribute, the “child”'s attribute value is taken (“parent” attribute value is overridden).

Notice that this mechanism works recorsivelly. It means that before the “parent” file is read, it is checked if there exists its “parent” (if yes, it is read).

3.4.6.2.3. Metadata file inheritance

This mechanism works in the same way as for `publication.properties` files with one exception—“parent” file should have `publication.rdf` name.

So, if we will specify in `publication.properties` file attribute publication.metadataFile, it will be checked if there exists its “parent” (`publication.rdf`).

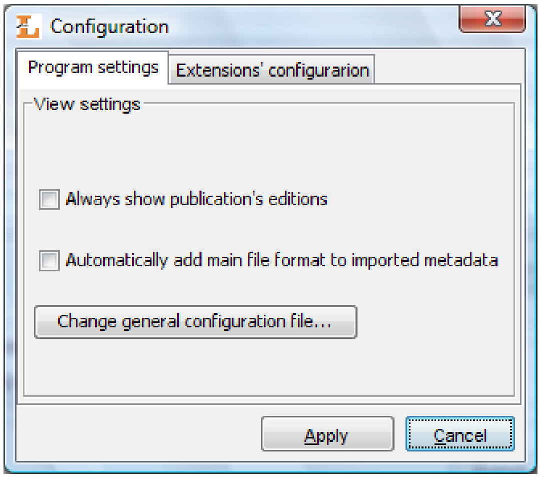
3.4.6.2.4. Examplary properties file

Menu item Save properties file... in menu Tools makes it possible to save information describing selected on the elements list publication (such a file can be used as an example for mass publication upload).

3.4.7. Editor's application configuration

After choosing the Configuration option from the Program menu configuration window will appear (Figure 3.40, “Application's configuration window”).

Figure 3.40. Application's configuration window


Configuration
Program settings
Extensions' configurarion
View settings
□ Always show publication's editions
□ Automatically add main file format to imported metadata
Change general configuration file...
Apply
Cancel

Tabs placed in configuration window allow user to configure particular application features. In the following sections all the tabs are described.

3.4.7.1. Program configurarion

Program settings tab enables user to adjust editor program view. This configuration is connected with editor program modes. If Always show publication's editions option *is not* checked then editor application works in simple mode, otherwise it works in advanced mode. Switching mode in editor application this option is switched in fact. The meaning of the option is described in Table 3.1, “Differences between editor's application work modes”.

Next configuration option is connected with Automatically add main file format to imported metadata check box. If this check box is selected then after metadata import process imported metadata are automatically enriched with edition main file format value. If this check box is not selected then imported metadata are not enriched.

Remaining configuration parameters may be retrieved from configuration file. In order to change configuration file select Configuration file... button and point on file or URL to configuration file. Please remember to select Use configuration from given source check box. The configuration file contains advanced configuration options. It is a properties file where keys and values are separated by equals character. The following parameters are in the configuration file:

- gui.lookandfeel - indicates name of the Java programming language class which is responsible for the way that GUI elements are displayed. If there is no value present then default GUI is used (basis on the OS).
- tags.refresh.period - the value indicates time in miliseconds between the update processes (retrieval of information from server) of tags list. Default value is 600000 which is 10 minutes.

- tags.message.period - the value indicates time in miliseconds between displaying messages that there are tags to moderate. Default value is 1800000 which is 30 minutes.

- newPublication.title.template - the value defines a template used to create title of an element (e.g. publication) which is being created. The template is a velocity template. In the template especially named variables are replaced with values from attributes. Those variable should have a name identical with RDF name of an attribute. Default value is `$!{Creator}#if ( ($Date || $Title) && $Creator ), #end$!Date#if ( $Title && $Date ), #end$!Title`. For example in place of `$Creator` variable value of an attribute which has *Creator* RDF name will be placed.

- searchDuplicates.roles.url - the value is a URL where the list of Dublin Core roles can be found. Attributes which have these roles should be used in duplicates search mechanism. Default value is http://fbc.pionier.net.pl/owoc/ajax.xml?id=GetDCRolesForDups.

- searchDuplicates.results.url - the value is the first part of a URL which will be used to perform duplicates search process. The second part state search parameters. Default value is http://fbc.pionier.net.pl/owoc/ajax.xml?id=GetDups.

- sortTokens - the value is a list of strings separated by `|` character. These strings are removed from the beggining of elements names when sorting elements on directories tree or elements list. Default value is `"|(|[...]||[|the |the`.

Examplary configuration file could have the following parameters (if there is no specific configuration key then its default value is used):

```
tags.refresh.period=60000
tags.message.period=3600000
sortTokens=[|]|/
```

3.4.7.2. Extensions' configuration

Extensions' configuration tab allows user to specify extensions' configuration. List of configurable extensions is placed on this tab (Figure 3.41, “Configuration - Extensions' configuration tab”). In order to change setting of specific configuration select extension from the list and press Configure... button. By default, the following extensions are available:

- MARC 21 communication format (configuration details can be found here).

- XML format (default configuration allows to import metadata from RDF and MASTER format, configuration details can be found here).

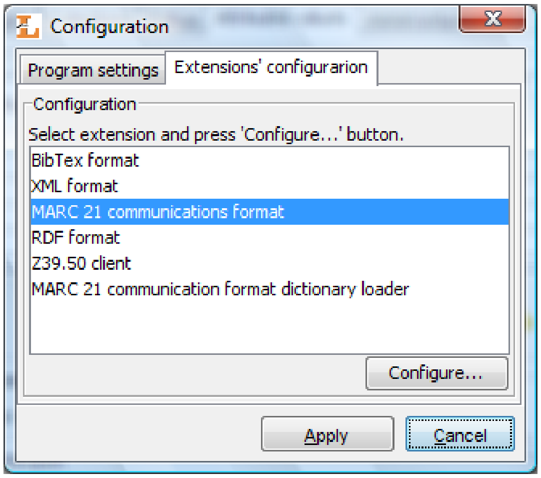
- Bibtex format (configuration details can be found here).

- RDF format (configuration details can be found here).

- Z39.50 extension (metadata retrieval, configuration details can be found here).

- Dictionary loader from MARC files (details can be found here).

Figure 3.41. Configuration - Extensions' configuration tab



Select extension and press 'Configure...' button.

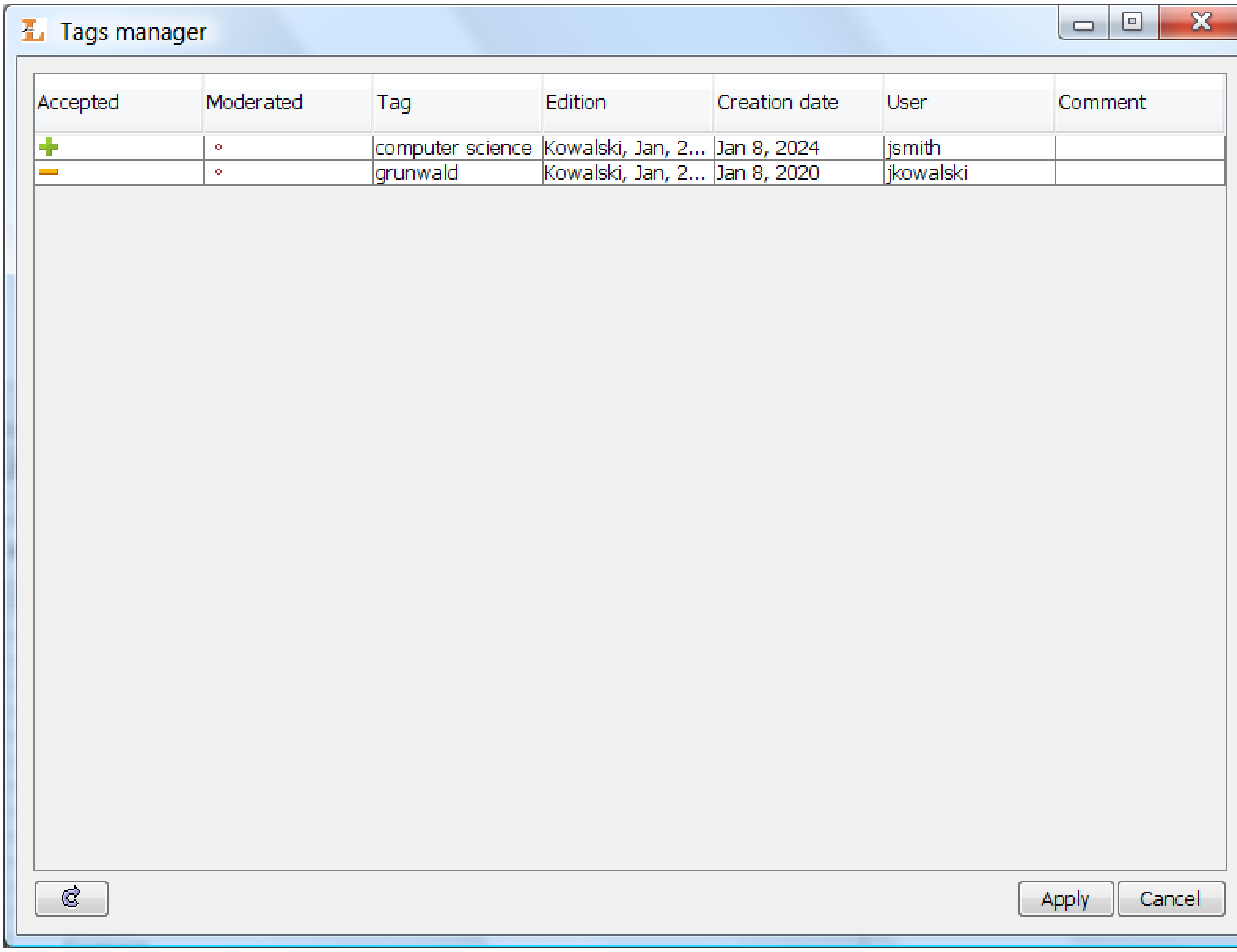
- BibTex format
- XML format
- MARC 21 communications format
- RDF format
- Z39.50 client
- MARC 21 communication format dictionary loader

3.5. Tags management

Web users of the *dLibra* system are allowed to provide tags (keywords) describing particular edition. Provided tags may be verified or modified by the editor or administrator of the *dLibra* system. Tag can be modified during specified in the configuration period (by default it is 30 days) from the moment web user adds it. After this period the tag is removed from the list of tag to moderate.

Tags to moderate management panel (Figure 3.42, “Tags management window”) is displayed in editor's application when Tags management... option from Tools menu is selected. On the panel there are all the tags to moderate (tags added during the last period; by default it is 30 days) which are assigned to the editions connected with logged editor.

Figure 3.42. Tags management window



Accepted	Moderated	Tag	Edition	Creation date	User	Comment
✚	◦	computer science	Kowalski, Jan, 2...	Jan 8, 2024	jsmith	
▬	◦	grunwald	Kowalski, Jan, 2...	Jan 8, 2020	jkowalski	

Tags on the management panel have the following properties (properties determine tag state):

- Accepted - two values are possible: ✚ icon indicates accepted tag (tag added to edition's metadata) and ▬ icon which indicates rejected tag (tag is not added present in edition's metadata). In order to change this value press the left mouse button on the “Accepted” column in a row corresponding to appropriate edition.

- Moderated - two values are possible: ◦ icon indicates moderated tag and ● icon indicates which is not moderated. If a tag is moderated it means that editor verified tag's properties and set its state. If a tag is not moderated it means that an editor did not see tag's properties. The state is automatically chanted to moderated when editor changes value of “Accepted” property. In order to change moderated value press the left mouse button on the “Moderated” column in a row corresponding to appropriate edition. Change of the value should generally concern the move to moderated value, e.g. when the editor has no objections concerning current tag state. Changing moderated value from moderated to not moderated should be rare as it is not natural order of changes.

- Tag - value added by the web user.
- Edition - name of the edition which tag was added to.
- Creation date - it is the date the web user added the tag.
- User - identifier (login) of the user which added tag.
- Comment - editior's comment concerning tag modification. The comment is presented to the web user which added the tag.

The list of tags is refreshed by default every 10 minutes, reminders for editor are displayed by default every 30 minutes. These periods can be modified according to the editor needs in application configuration. In addition, in the bottom-right corner of the management panel Refresh button is located. This button should be used to refresh list of tags without the need to wait for automatic refresh.

In order to apply all changes made on the tags management panel press Apply button. In order to reject made changes press Cancel button.

In *dLibra* system two initial tag states are possible:

- Accepted tag - by defaul each tag is accepted therefore it is added to edition's metadata. If the editor will not chage the tag state during the period defined in configuration (by default 30 days) then it will be removed from the list of tags to moderate and it will be left in the edition's metadata.
- Rejected tag - by default each tag is rejected. If the editor will not change the tag state during the period defined in configuration (by default 30 days) then it will be removed from the system (the tag will not be present in edtion's metadata).

In order to set appropriate initial state for all new tags contact *dLibra* system administrator.

3.6. Publication Creator

This chapter contains information about Publication Creator, which has been added as a part of Editor's Application.

3.6.1. Publication Creator's functions

Publication Creator generates WWW pages in HTML format basis on the files provided by the user. WWW pages contain references to these file or/and the content of these files (e.g. pictures). Publication Creator is useful when user wants to create a compact document containing files they have (for example picture gallery) in a simple, quick way. Creator fills document templates with files (in case of pictures it prepares also miniatures) and saves as a final publication on a local user's disk. After publication is generated *dLibra* system asks if the user wants to add this publication to the library. If yes, new publication wizard is started. In the first step user selects destination (either directory or group publication). The remaining steps are the same as for new publication wizard.

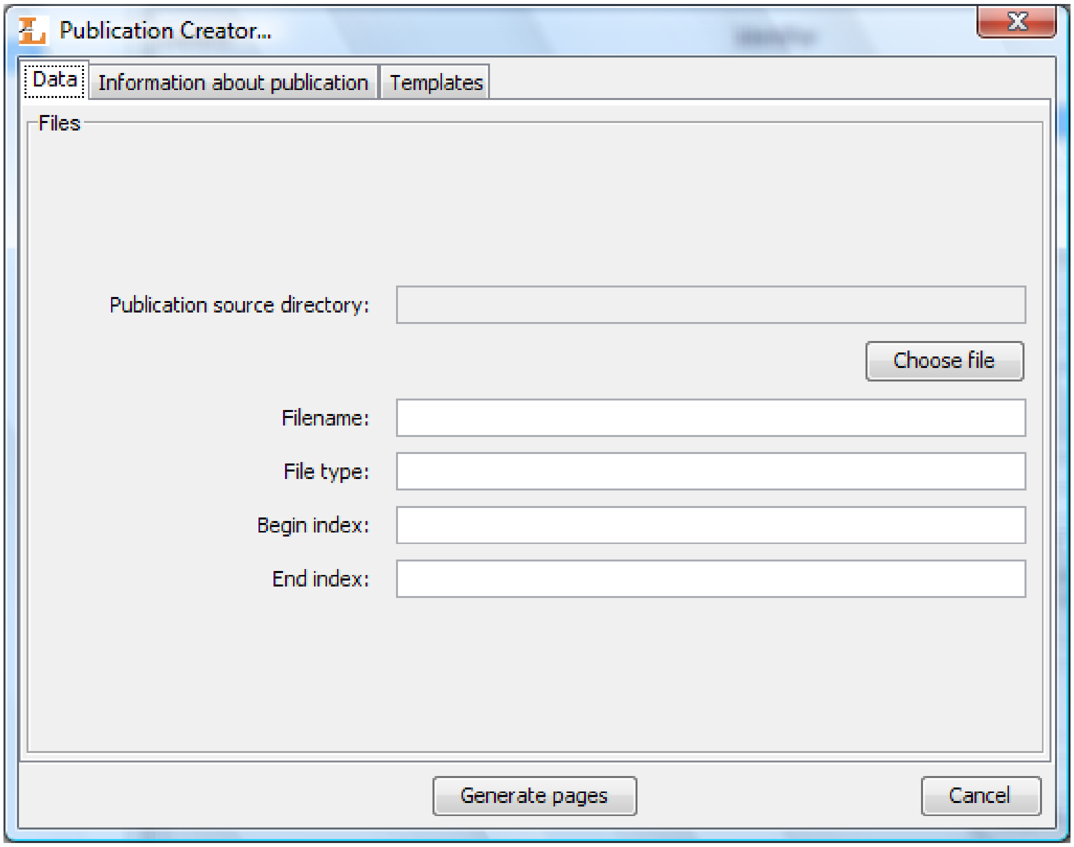
Before the publication is generated, all files must be properly prepared. They all must be placed in the same directory on a local user disk. Furthermore the files must have the precise format of name. Base segment of a name must be the same in all files, they must be differentiated only by the ordinal number placed just before the file extension (for example: `test01.jpg`, `test02.jpg`, `test03.jpg` etc.).

3.6.2. Using Publication Creator

Publication Creator is composed of three tabs: Data, Information about publication and Templates. If Generate pages button is pressed application generates publication files in the directory chosen by the user. Cancel button closes the window.

3.6.2.1. "Data" tab

Figure 3.43. Publication Creator - Information about files tab



On the Data tab user may choose files to use when generating publication files. Choose file button allows user to select one of the files which will be used in generation process. When the file is selected fields in a File panel are filled with values:

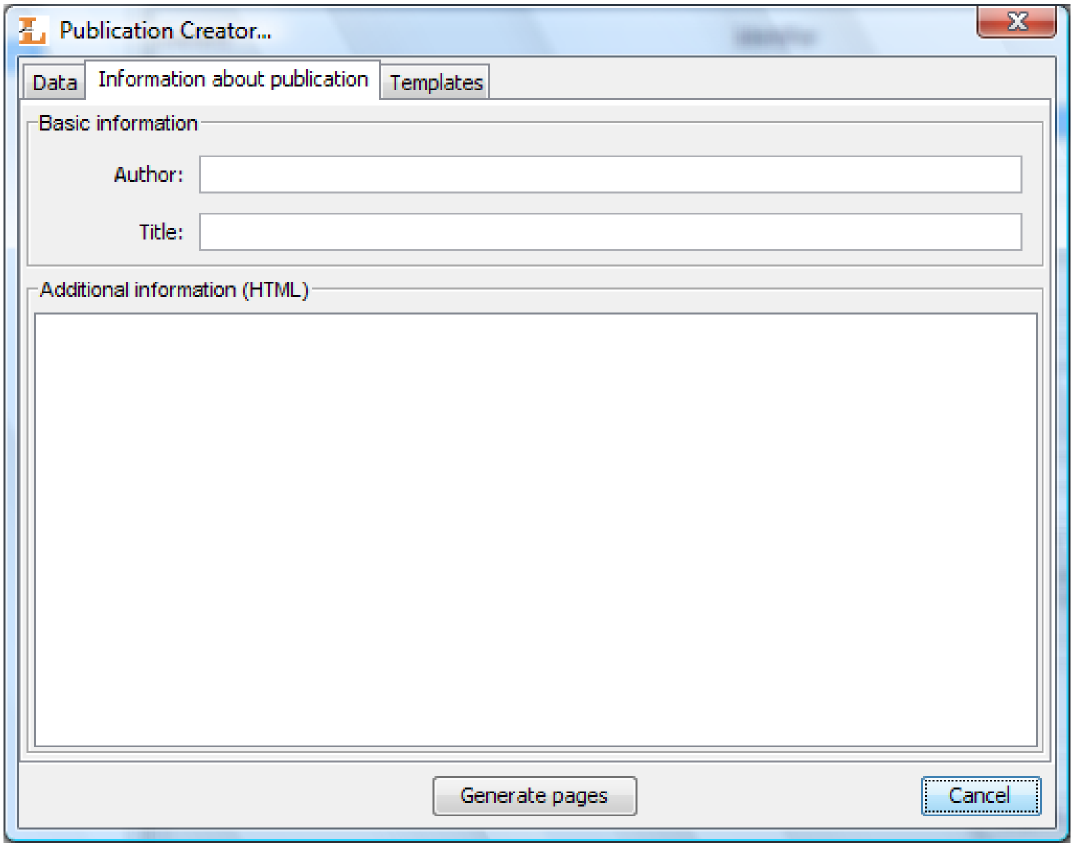
- Filename - base segment of files name,
- File type - files' format,
- Begin index - the number of the first file that will be used to generate a publication,
- End index - the number of the last file that will be used to generate a publication.

These values may be edited by the user. Particularly changing the index values may be useful if publication is not meant to contain all images placed in a chosen directory.

Publication will be generated to directory where the selected files are located (this directory is pointed by Publication source directory field). Next, the application will ask the user if he/she wants to submit the publication to the system. If the answer is positive then appropriate wizard will be started. In the first step it is necessary to select the element where new publication will be placed, next steps are the same as in new publication wizard.

3.6.2.2. "Information about publication" tab

Figure 3.44. Publication Creator - Information about publication tab

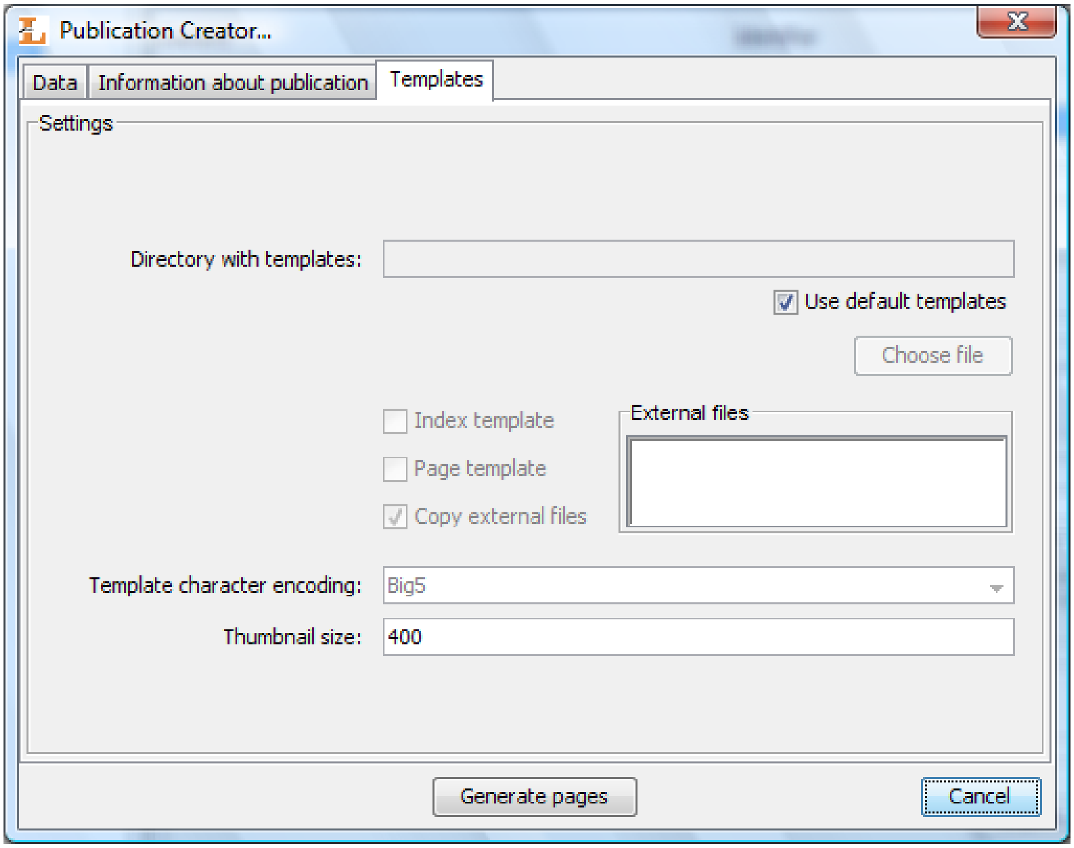


Additional information concerning the publication may be provided on the Information about publication tab. The information will be placed in the HTML files and include:

- Author - Information about publication's author
- Title - A title of the publication
- Additional information - Additional information about the publication (user may use here HTML tags).

3.6.2.3. "Templates" tab

Figure 3.45. Publication Creator - Information about templates tab



Directory with templates:

☑ Use default templates

□ Index template

□ Page template

☑ Copy external files

Template character encoding: Big5

Thumbnail size: 400

To use default configuration, simply check Use default templates.

The Templates tab allows user to select templates which are to be used to generate a publication. To choose templates, press Choose directory button and choose directory with templates. Index template file should have an `index.vm` name, and page template file should have a `page.vm` name. External files list contains all additional files that are used by the templates. If Copy external files is selected then after generation process external files will be copied to target publication directory. The user may also select template character encoding and size of thumbnails which will be generated (thumbnail are only generated when the publication is based on graphics).

Templates should use the Velocity standard to use to following variables:

- $info.PublicationAuthor - information about publication author,
- $info.PublicationTitle - publication title,
- $info.PublicationInfo - additional information about publication,
- $relPath - relative path to the publication sources,
- $info.PictureName - file name (base segment),
- $info.PictureType - file type (file name extension),

- $pictureNumber - file number,
- $info.BeginIndex - index of first file,
- $info.EndIndex - index of last file,
- $info.PageCount - the number of files in the publication,
- $pageNumber - actual page number,
- $info.PageName - name of the page file,

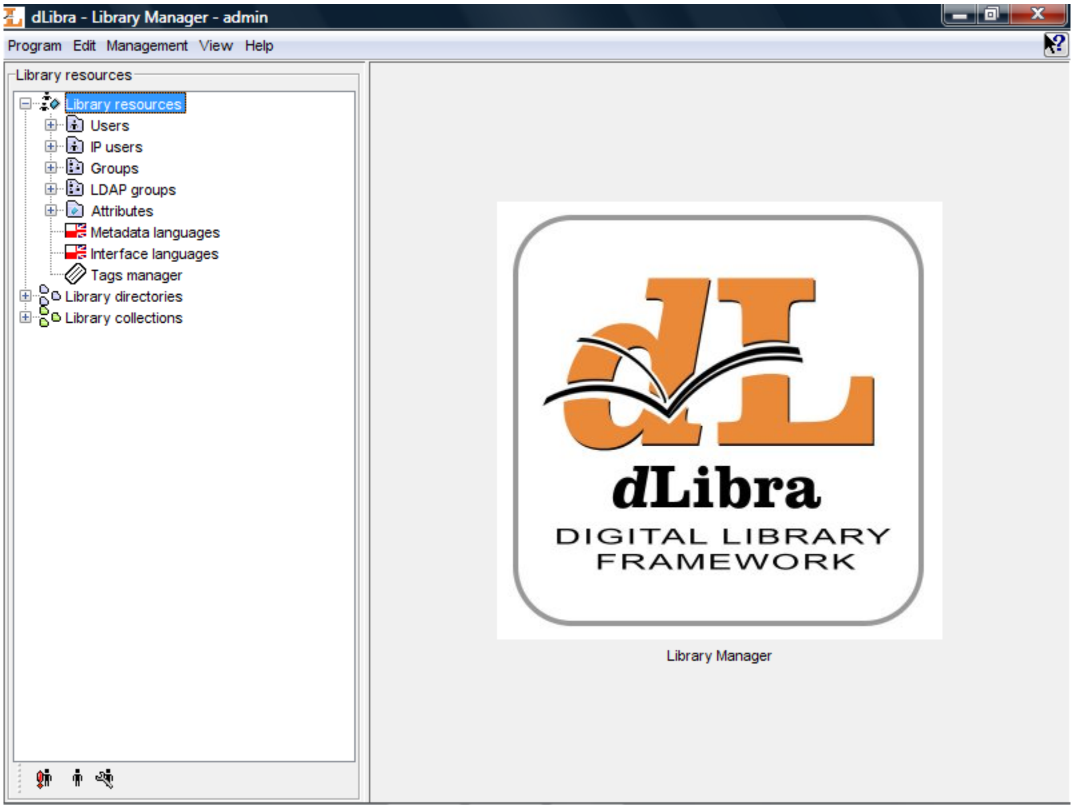
For more information about Velocity format visit Velocity homepage: http://jakarta.apache.org/velocity/user-guide.html.

Chapter 4. Library Manager

This chapter describes the *dLibra* Administrator's Application with regard to the most common library management tasks.

4.1. Administrator's Application main window

Figure 4.1. Administrator's Application main window



The main window of the Administrator's Application is divided into two parts. The Library Tree, which is located on the left, provides access to library resources such as library users, groups and attribute schemes as well as to the directory of library content. On the right, property dialogs are displayed accordingly to the element selected in the library tree.

4.1.1. Library Tree

The Library Tree is divided into three subtrees: resources tree, directories tree, and collections tree.

Resources tree contains nodes of the following types:

- users node - contains nodes representing library users

- user node - represents single library user.
- groups node - contains nodes representing user groups.
- group node - represents single user group.
- LDAP groups node - contains nodes representing LDAP group nodes - visible only whe *dLibra* is configured in such a way that it uses LDAP server
- LDAP group node - represents one LDAP group
- attributes node - contains nodes representing attributes defined in the library.
- attribute node - represents single attribute.
- languages management - allows administrator to manage lenguages available in the system.
- tags management - allows administrator to manage all tags added by web users.

Apart from the node corresponding to library resources, the Library Tree contains a node providing access to the library content. The structure of the node is explained in detail in Editor's Application. The functionality is though limited only to rights management for all tree objects.

“Directories Tree” is placed in a Library collections node. It contains nodes of only one type - collection node, which represents library collection. Collections are grouped in a library in hierarchical structure. Each collection may contain a number of subcollections. On every level of this structure a collection may be assigned a publication. Any number of publications may belong to a single collection. If publication belongs to a collection it implies that it also belongs to all its supercollections on the path to main library collection. A context menu is associated with collection node. It provides quick access to three operations: adding, deleteing and refreshing a collection. Context menu is called by clicking a right mouse button on a node. When we select collection on the properties window we can modify properties (???) of the collection, assign publications (???) and rights (Section 4.3.4, “Collection-level permissions”) to it.

With every node on library tree a context menu is associated that provides quick access to the most common operations performed on a given type of element (e.g. creating a new directory).

Similar to Directories Tree (see Editor's Application) it is possible to hide Library Tree nodes. Currently, it is possible to hide users nodes.

4.2. Basic administrator's tasks

This section describes how to complete common administrator's tasks using the *dLibra* Administrator's Application.

4.2.1. Creating/removing collection

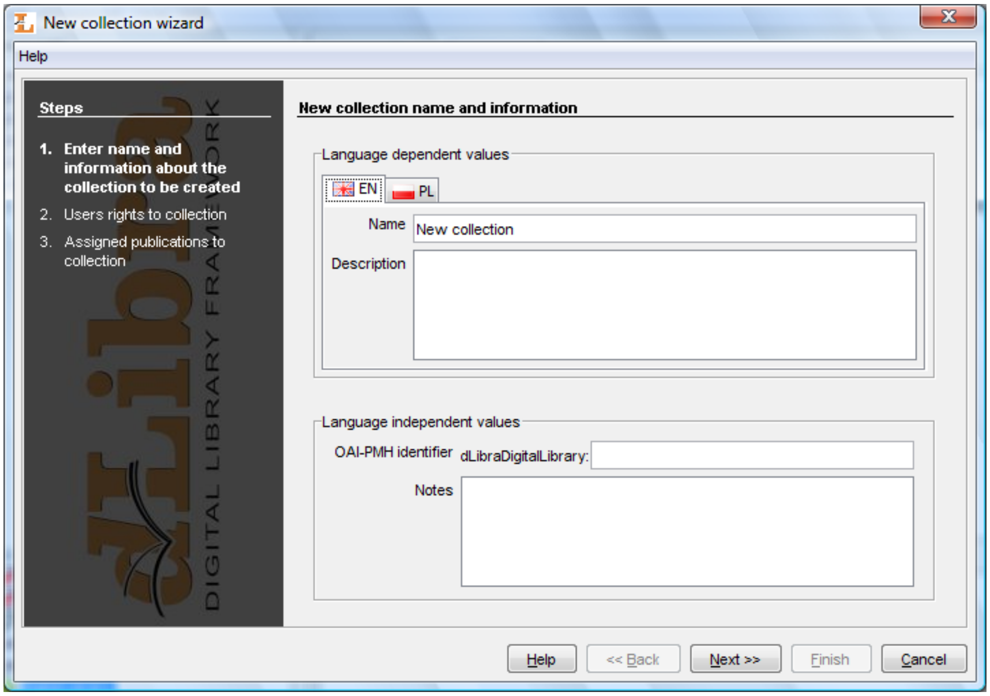
To create a new collection:

1. In the Library Tree select the library collection which will be the supercollection for a newly created one.

2. From the popup menu select the New collection... function. Alternatively, the toolbar button or main menu options can be used.

3. In the first step enter name, description and notes for the new collection. Press Next button to proceed to the next step.

Figure 4.2. New collection wizard - general information



New collection wizard

Steps

1. Enter name and information about the collection to be created
2. Users rights to collection
3. Assigned publications to collection

New collection name and information

Language dependent values

EN | PL

Name: New collection

Description:

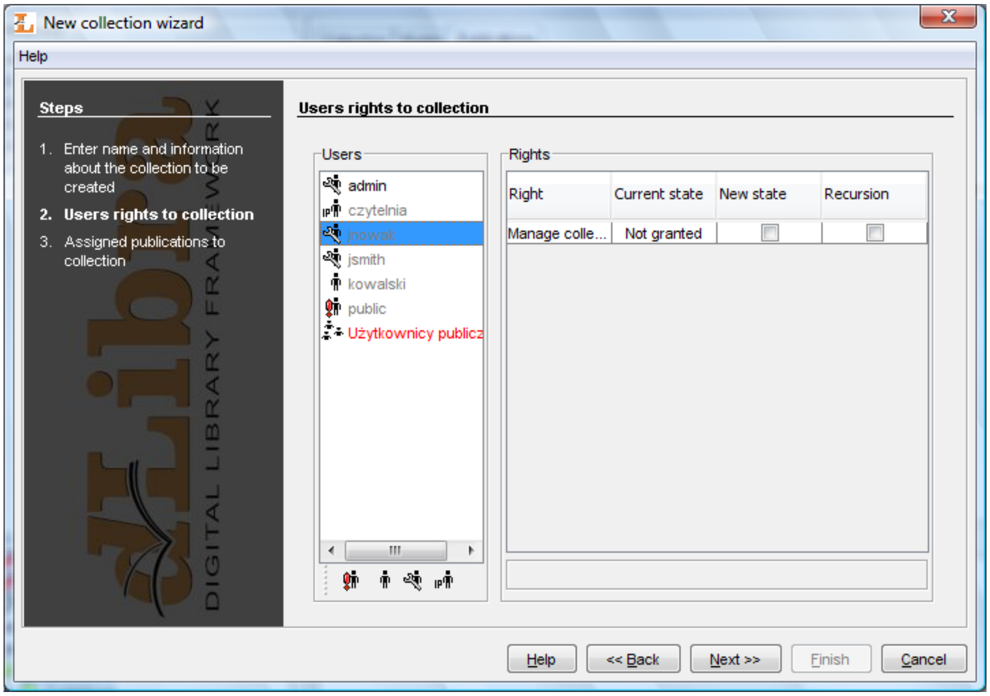
Language independent values

OAI-PMH identifier: dLibraDigitalLibrary:

Notes:

4. In this step grant appropriate rights to users. Press Next button to proceed to the next step.

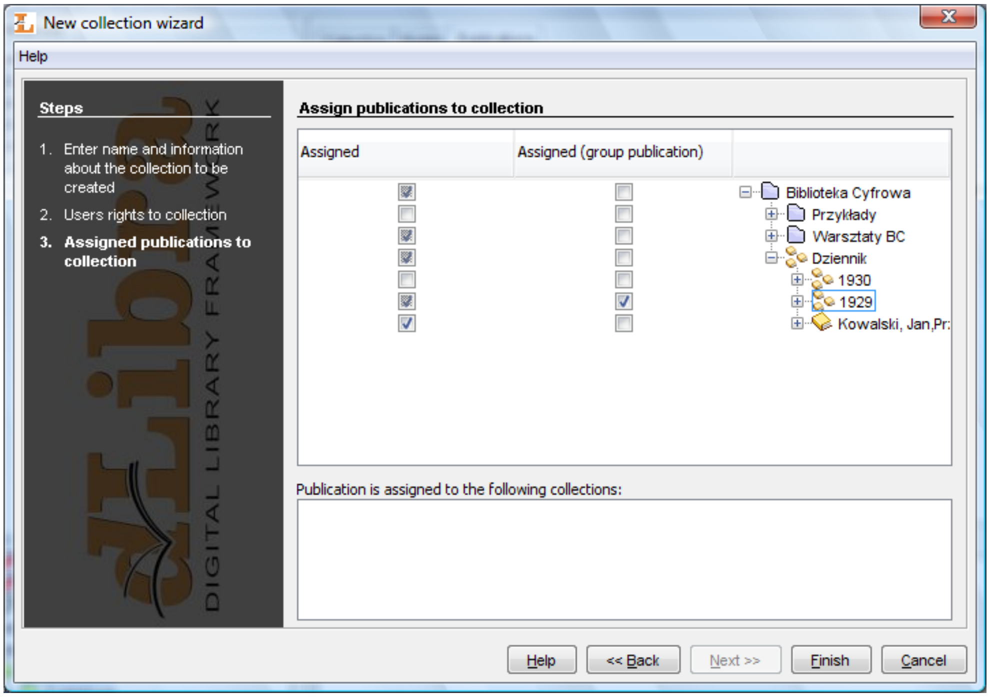
Figure 4.3. New collection wizard - rights


New collection wizard
Help
Steps
1. Enter name and information about the collection to be created
2. Users rights to collection
3. Assigned publications to collection
DIGITAL LIBRARY FRAMEWORK
Users rights to collection
Users
admin
czytelnia
jnowak
jsmith
kowalski
public
Użytkownicy publicz
Rights
Help
<< Back
Next >>
Finish
Cancel

Right	Current state	New state	Recursion
Manage colle...	Not granted	□	□

5. In the last step the user may assign publications to new collection. Press Finish button to create new collection.

Figure 4.4. New collection wizard - publications


New collection wizard
Help
Steps
1. Enter name and information about the collection to be created
2. Users rights to collection
3. Assigned publications to collection
DIGITAL LIBRARY FRAMEWORK
Assign publications to collection
Publication is assigned to the following collections:
Help
<< Back
Next >>
Finish
Cancel

Assigned	Assigned (group publication)	
☑	□	Biblioteka Cyfrowa
□	□	Przykłady
☑	□	Warsztaty BC
☑	□	Dziennik
□	□	1930
☑	☑	1929
☑	□	Kowalski, Jan,Pr:

To remove a collection:

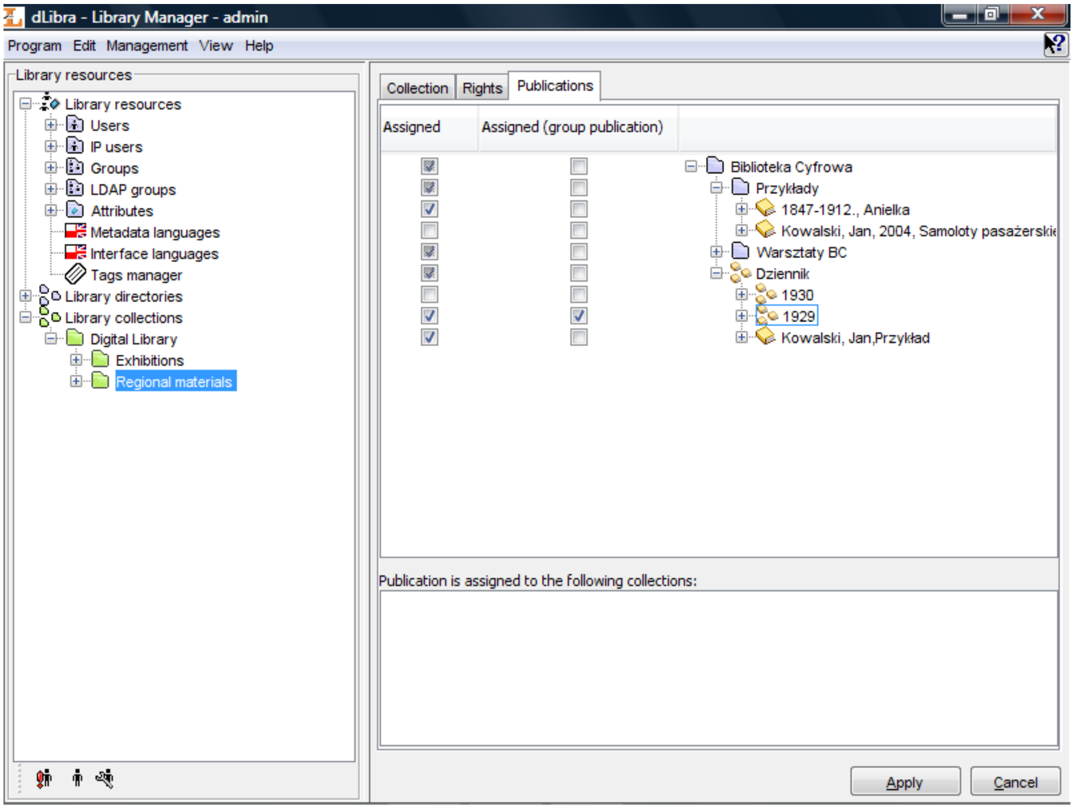
1. In the Library Tree Select the collection to be removed.

2. Press the 🗑 Delete button. Alternatively, a context menu or a pull-down menu can be used.

4.2.2. Managing publications in collection

There is a possibility to manage publications membership for a given collection. A user who has a right to manage a given collection can add publications to it and remove publications from it. An interface enabling it is placed on a tab named Publications available after choosing a certain collection in “Collections Tree”.

Figure 4.5. Collection's publications management



After choosing certain collection in “Collections Tree”, on the tab named Publications (Figure 4.5, “Collection's publications management”) appears an interface enabling publications' membership management (for selected on “Collection Trees” collection).

On the top of the Publications tab a table used for editing publications' to collection assignment is placed. The table has two columns. In the right column library structure is placed (the same as in “Directories Tree” but not editable). An element in the right column (in “Directories Tree”) is selected or not when corresponding check box on the left side is selected or not. There are three possible states of this check box:

- □ - state possible for every element. In case of a directory it means that none of the publications which it encloses is assigned to selected collection. In case of a publication it means that the publication is not assigned to selected collection.

- ▨ - state possible only for directories and it means that a directory encloses both publication(s) which are assigned to selected collection and publication(s) which are not assigned to selected collection,

- ☑ - state possible for every element. In case of directory it means that all publications in this directory are assigned to selected collection. In case of publication it means that the publication is assigned to selected collection.

Modifying check boxes publications' assignment to collection is modified.

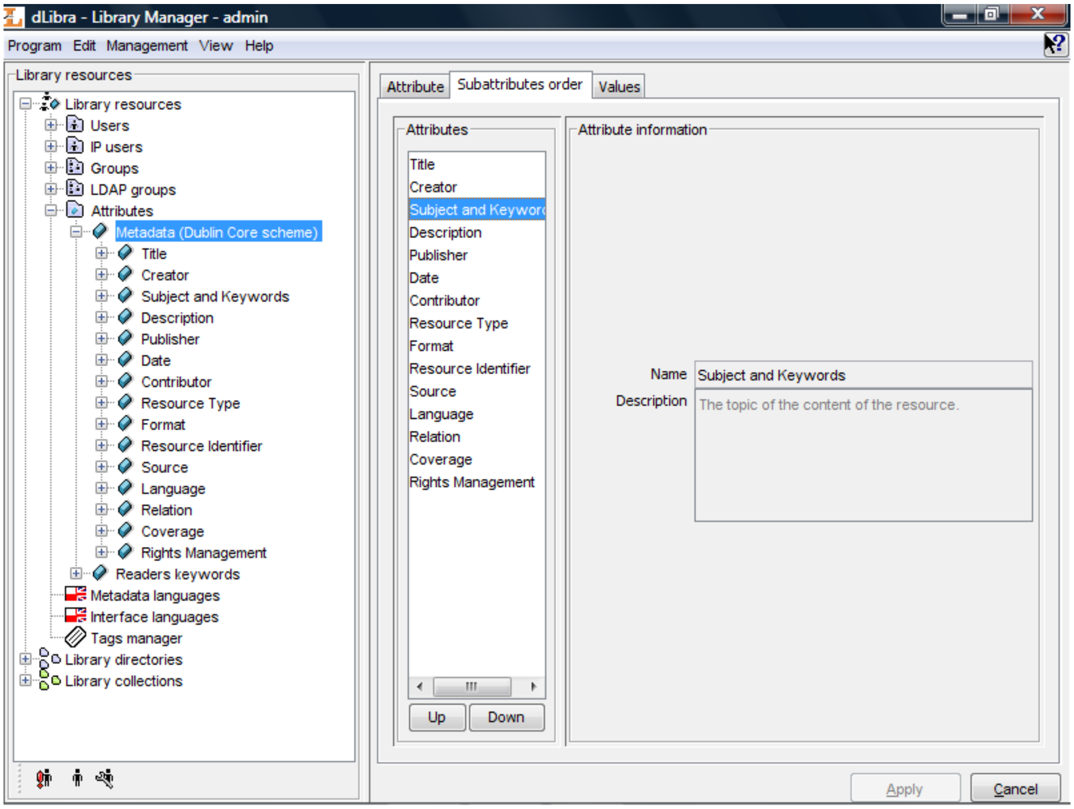
All changes have to be approved by pressing Apply button on the bottom right of Properties Window.

4.2.3. Library attributes management

Library Manager application allows user to manage attributes in *dLibra* system. It is possible through interfaces available after choosing attributes node or single attribute node in Library Tree.

Attributes in *dLibra* system are stored in a certain order, which is important in view of presenting bibliographic description of library resources. For every attribute (and the attributes node) it is possible to manage the order of its subattributes on the Attributes order tab (Figure 4.6, “Attributes order management”).

Figure 4.6. Attributes order management



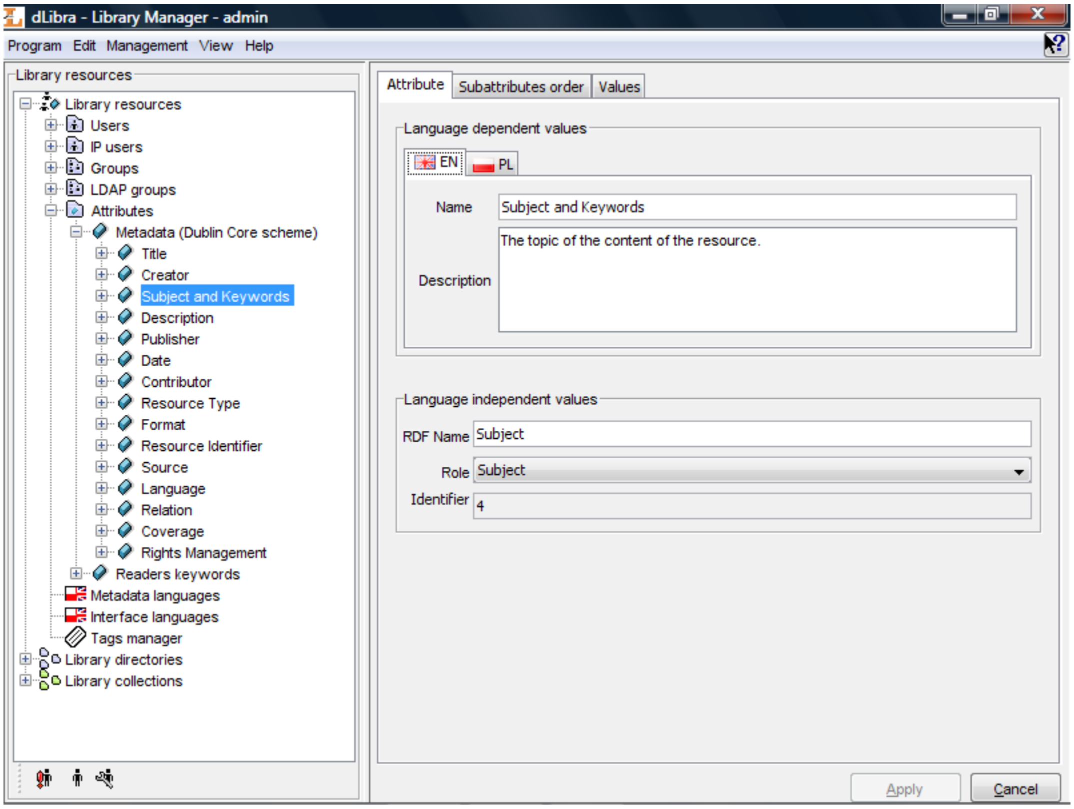
Name: Subject and Keywords

Description: The topic of the content of the resource.

On the left side there is a list of attributes and on the right side there is information about currently chosen attribute. These values are only informational and cannot be edited here. Order of attributes in the list can be easily changed using Up and Down buttons. All changes must be confirmed by pressing Apply button down the screen.

Editing attributes properties can be performed using the interface placed on a General information tab available after choosing single attribute node in the Library Tree. User can change name, RDF name, role and description of each attribute.

Figure 4.7. Attribute properties



Name: Subject and Keywords

Description: The topic of the content of the resource.

RDF Name: Subject

Role: Subject

Identifier: 4

Name and description of an attribute are multilingual properties and they have informational character. RDF name is a unique name identifying attribute inside Dublin Core metadata scheme and outside the system. A single role can be assigned to an attribute, if it has a special meaning in a system and it's value is used in other contexts. *dLibra* system lets user to assign attribute one of the following roles:

- Contributor
- Coverage
- Creator
- Date
- Description
- Format

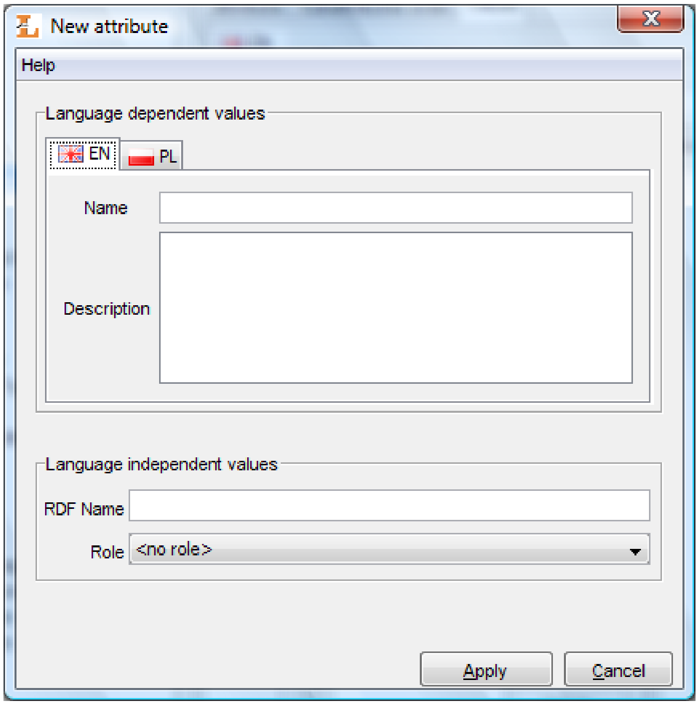
- Language
- Publisher
- Relation
- Resource identifier
- Resource type
- Rights management
- Source
- Subject and keywords
- Title
- Tags - accepted tags will be added to values of the attribute which has *Tags* role.

4.2.3.1. Adding and removing attributes

To add new attribute:

1. Choose attributes node in the Library Tree. An interface to manage attributes will appear on a tab Attributes management.
2. Click Add button placed below the attributes list on the left.
3. In the New Attribute Wizard enter name and description for an attribute for all languages. Enter the RDF name - a unique attribute identifier To create new attribute press Apply button.

Figure 4.8. New Attribute Wizard



Newly created attribute will appear as the last element in attributes list and will be "grayed". Unless changes are committed by clicking Apply button there is no way to edit it or change it's order. After applying it's colour changes to black and attribute is fully functional.

To remove attribute:

1. In the Library Tree select the attribute's node to remove.
2. Click the right mouse button on it and select Delete option. If the attribute is connected with values then system will ask for confirmation, if not the attribute will be removed.

4.2.3.2. Attributes editing

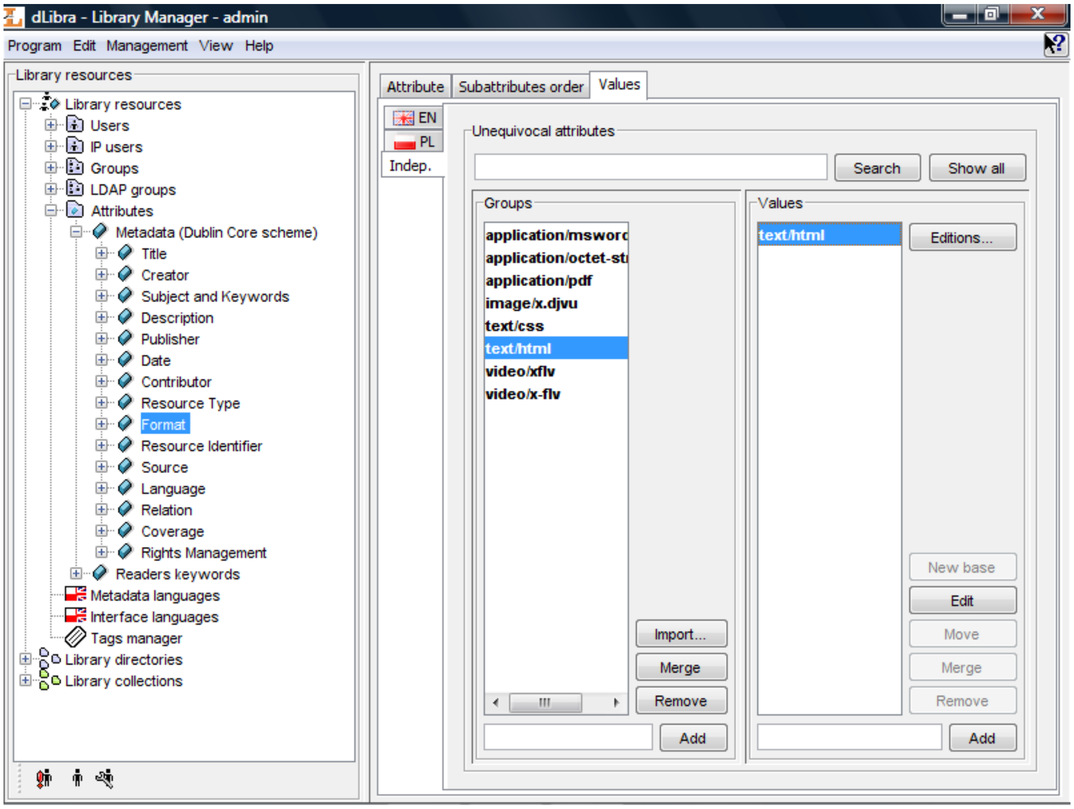
To edit properties of an attribute:

1. Choose single attribute node in the Library Tree. An interface to manage attributes' properties will appear on a tab General information.
2. Enter or choose new values in the fields Name, RDF Name, Role and Description. It is worth remembering that name and description are multilingual values. If a role is chosen that some other attribute has already had assigned, user will be asked to confirm the operation. Assigning this role to a new attribute will cause old attribute to be assigned "no role" value.
3. All changes must be committed by clicking Apply button.

4.2.4. Attribute's values dictionary management

The following section describes in details functions and operations allowing the management of synonyms' dictionary and describes steps to be taken to commit these actions.

Figure 4.9. Attributes values management



Interface to manage attributes' values is available after choosing a single attribute node in Library Tree and a tab named Values. Panel is divided into two parts - list o synonyms' groups on the left and list of single group values on the right. List of values is filled with values after choosing single group on the left list. Each list has additionally a set of buttons allowing to perform some operations on dictionary. All actions are described in details later in this section.

4.2.4.1. Managing of synonyms' groups

To create a new synonyms' group

1. In a text field placed below the groups list, a value for a base value in a new group should be entered. This value will also become a name for a new group. Value must not be already in use by some other group

2. To create a new group click Add button placed beside the groups list.

To remove a group of synonyms:

1. Choose a single group on a groups' list.

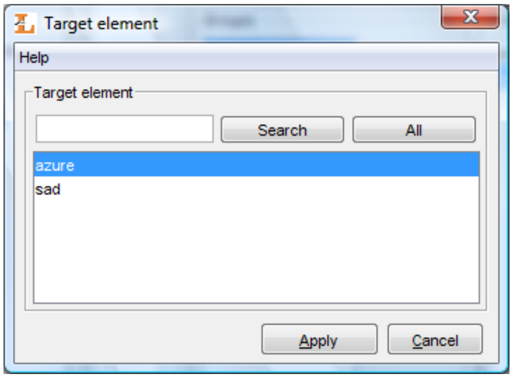
2. Press Remove button placed beside the groups' list. After confirming, a group and all it's values will be removed from the list and from the system.

   Caution! If any of the group's values is assigned to a bibliographic description of some library element, the system won't delete such a group and will show a dialog with information that all connections should be removed instead.

To merge two groups of synonyms:

1. Choose a group in a list, that has to be merged with other group.
2. A button Merge should be pressed. After confirming an intention of merging the groups a new window with groups' list will appear.
3. From a new list (Figure 4.10, “Choosing a group to merge”) choose a group that previously chosen group should be merged to. Edition field up the window makes it easier to find a groups with names containing certain set of characters. An All button brings all the groups back to the list.

Figure 4.10. Choosing a group to merge



4. To finish the process of merging groups press Apply button. Previously chosen group will be merged with new one and it means that old group will be removed and all it's values will be moved to a new one.

   Caution! If the target group contains a value with the same name as one of the values from group chosen to be merged, system won't merge these groups and instead will show a dialog with information that this operation cannot be proceeded. Before merging groups it must be assured that there are no duplicate values' names in groups.

4.2.4.2. Managing the attributes' values

All actions can be performed after choosing a certain group in the left list.

To add a value to a group:

1. In a text field placed below the values' list enter the name for a value that should be added to a group. Name cannot be already in use by other value in a group.
2. In order to add a value press an Add button placed beside the text field.

To remove a value from a group:

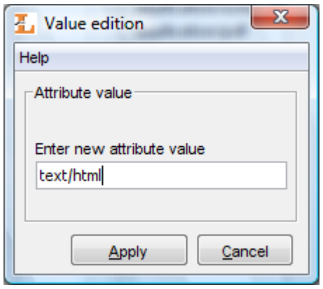
1. Choose a value from the right list that has to be removed.
2. Press the Remove button placed beside the list. Value will be removed from the list and from the system.

   Caution! If the chosen value is assigned to a bibliographic description of some library element, the system won't delete such a value and will show a dialog with information that all connections should be removed instead.

To change the value's name:

1. Choose a value which name should be changed.
2. Press the Edit button. A dialog allowing to edit a value will appear.
3. Enter a new name for a value and press Apply button. If the group's base value was edited then the group will get a new name as well.

Figure 4.11. Editing an attribute's value's name



Enter new attribute value: text/html

To merge two values in a group:

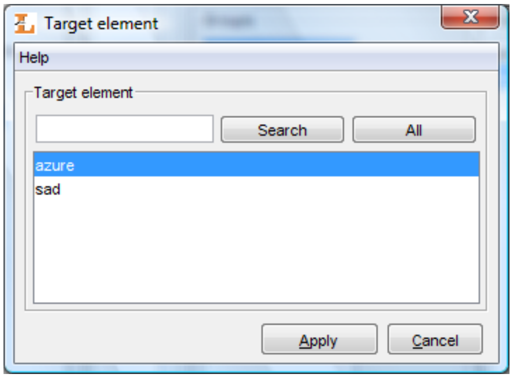
1. Choose the value that should be merged with other value.
2. Press the Merge button placed beside the values' list. After confirming a new window will appear with the values' list.
3. From a new list choose a value to which previously chosen value should be merged to. Edition field up the window makes it easier to find the values with names containing certain set of characters. An All button brings all the values back to the list.
4. To finish the operation of merging the values press Apply button. Previously chosen value will be merged with new one which means that all connections to it will be removed with connections to a new one.

To move a value to a new group:

1. Choose a value that has to be moved to a new group.
2. Press the Move button. After confirming a new window will appear with the groups' list that this value may be moved to.
3. From a new list choose a group to which previously chosen value should be moved to. Edition field up the window makes it easier to find the groups with names con-

taining certain set of characters. An All button brings all the groups back to the list.

Figure 4.12. Attribute's name edition dialog



4. To finish the operation of moving the value press Apply button. Previously chosen value will be moved to a new group.

   Caution! If the chosen group already contain a value with the same name, system won't move the value to that group and will show a dialog with information that this operation cannot be performed instead.

To change base value in a group:

1. Choose a value form the list that has to be the new base value.

2. Press New base button. Chosen value will become a new base value in a group and therefore a group's name will change into a new one.

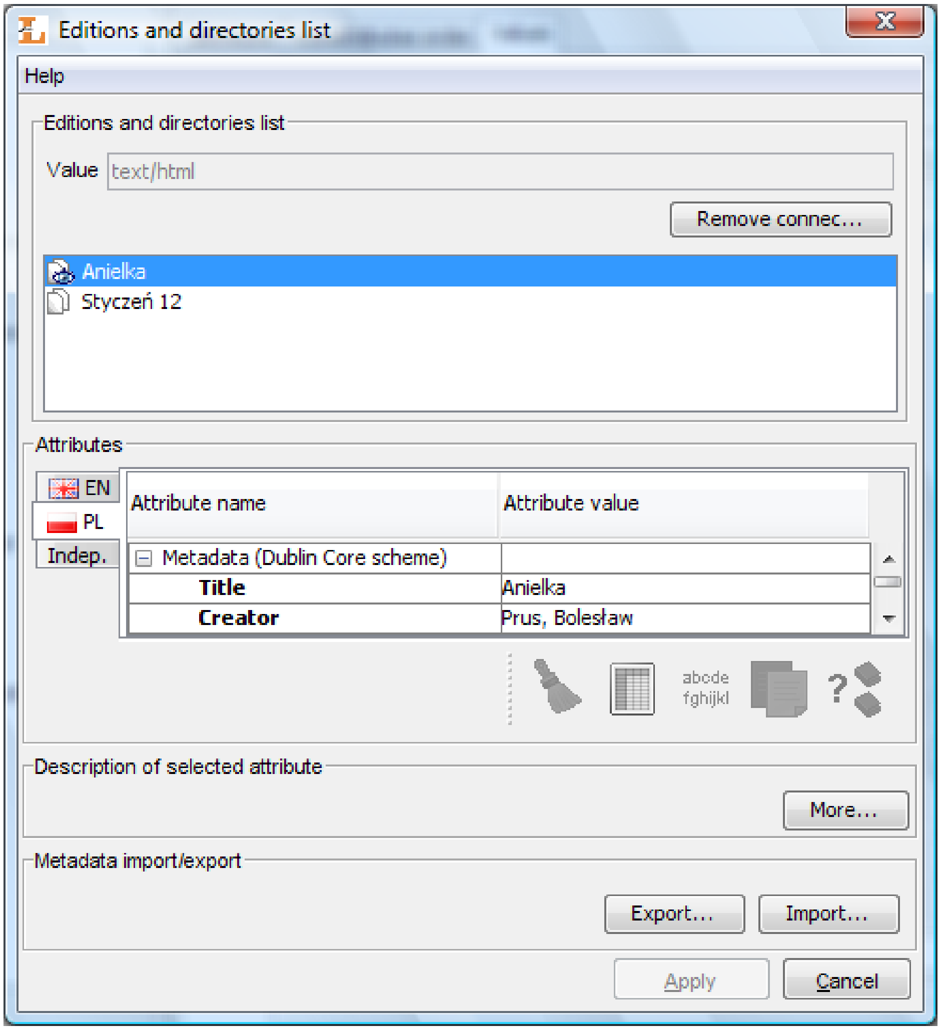
   Caution! Be sure that there is no group with the same name as chosen value. If such a group already exist the operation of changing base value will not finish correctly.

To remove the connections of a value:

1. Choose a value from a list which connections should be removed.

2. Press the Editions... button. A new window will appear containing a list of editions and directories that bibliographic description is connected with the value. Pressing Cancel button any time will cancel all changes made.

Figure 4.13. List of editions and directories connected with a value



Value: text/html

Anielka
Styczeń 12

Attribute name	Attribute value
⊟ Metadata (Dublin Core scheme)	
Title	Anielka
Creator	Prus, Bolesław

3. Press the Remove connections button. All editions and directories will be removed from the list.

   Caution! In case user has no rights to modify any of the editions or directories, a dialog will appear informing about the problem and no connections will be removed.

4. Press the Apply button to finish the process of removing the connections.

4.2.4.3. Importing groups of values into the dictionary

Import... button on the dictionary management panel allows editor or administrator to import groups of values from external sources. The mechanism which imports groups of values is based on the extensions which provide such functionality. By default, *dLibra* system includes the following extensions which import groups of values int the dictionary:

- MARC groups of values import - details can be found here.

If there is only one such an extension in the application then the Import... button invokes the extension and the user should act according to description of the extensions. If there are more extensions then firstly user selects extension which should be used and then acts according to description of the extension he has chosen.

4.2.5. Users and groups management

4.2.5.1. Library users management

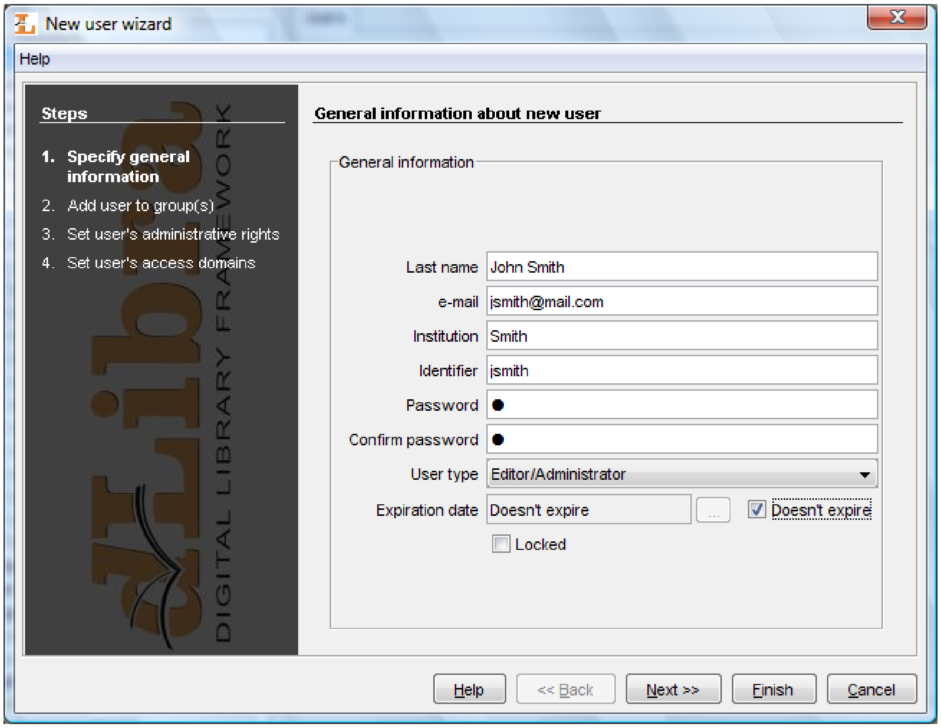
To create a new library user:

1. In the Library Tree select the users node or a user node.
2. From the popup menu select the New user... option. Alternativelly choose ✚ button from toolbar or opcję New user... from Management menu.
3. In the first step (Figure 4.14, “New user wizard - general information”) provide user's general information. Identifier, e-mail address and password are required. Choose one of the three user types:

   - Administrator/editor - dLibra system user, can log in to any application (including editor and administrator application), usually his/her task is administration of the *dLibra* system or library content creation
   - Reader - WWW user, cannot log in to editor and administrator application, can log in on WWW pages and modify information concerning his/her account,
   - Reader with restrictions - has the same possibilities as the Reader excluding possibility to modify his/her account.

   To go to the next step, press the Next button. It is to possible to create the user here by pressing the Finish button.

Figure 4.14. New user wizard - general information



Steps

1. Specify general information
2. Add user to group(s)
3. Set user's administrative rights
4. Set user's access domains

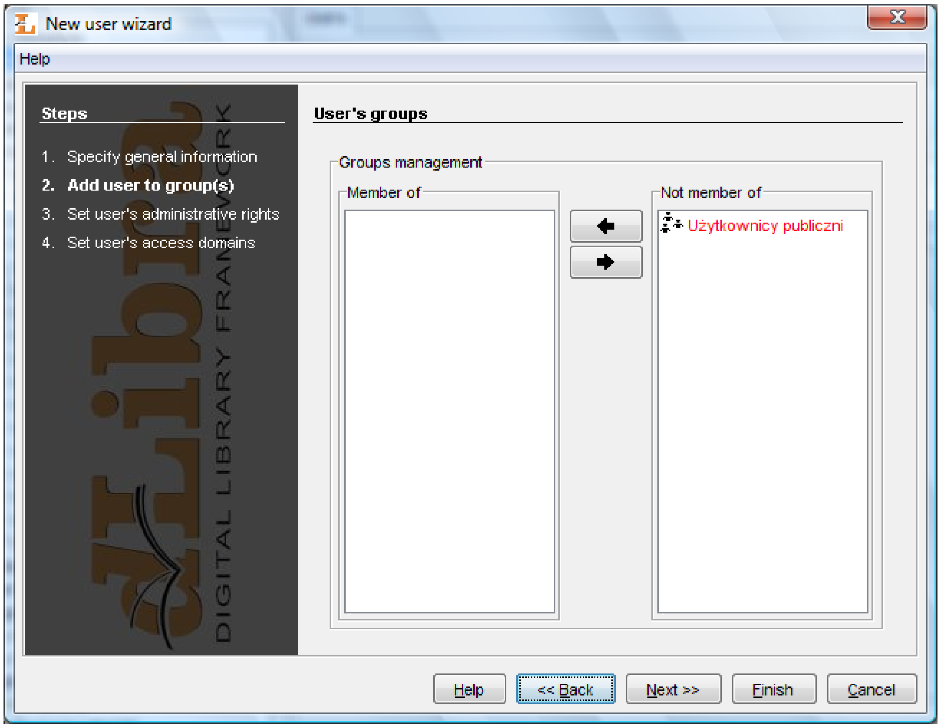
General information about new user

General information

Last name	John Smith
e-mail	jsmith@mail.com
Institution	Smith
Identifier	jsmith
Password	●
Confirm password	●
User type	Editor/Administrator
Expiration date	Doesn't expire ... ☑ Doesn't expire
	□ Locked

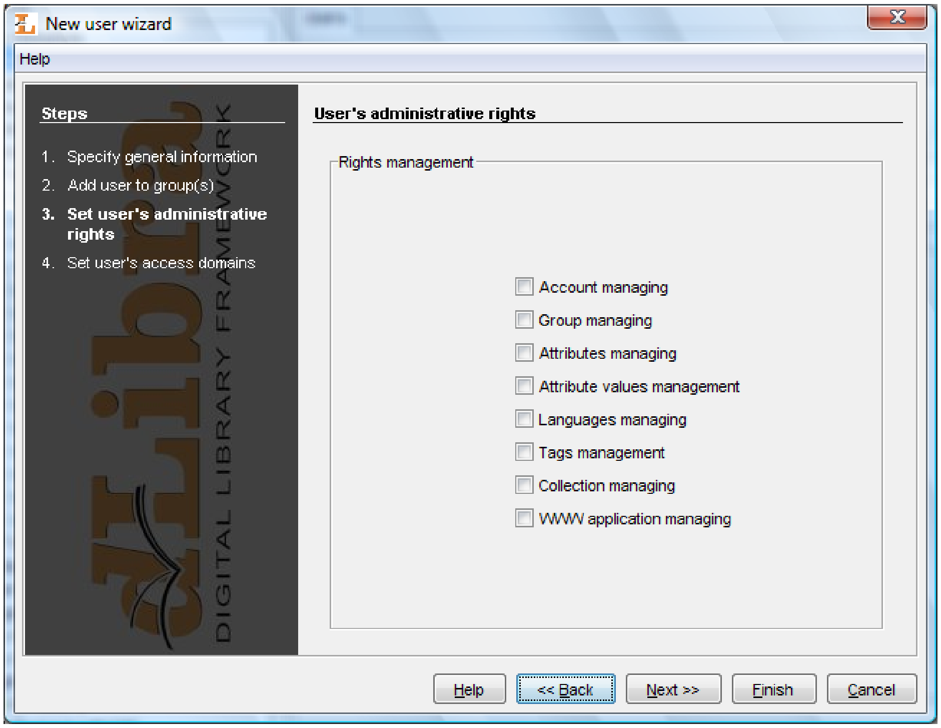
4. In the second step (Figure 4.15, “New user wizard - assigning to groups”) assign user to groups. Member of list constains groups which the user will assigned to, Not member of list contains groups which the user will not assigned to. Press the Next button to go to the next step.

Figure 4.15. New user wizard - assigning to groups



5. In the third step (Figure 4.16, “New user wizard - assigning administrative rights”) assign administrative rights to user (the meaning of rights is described in ???). Press the Next button to go to the next step.

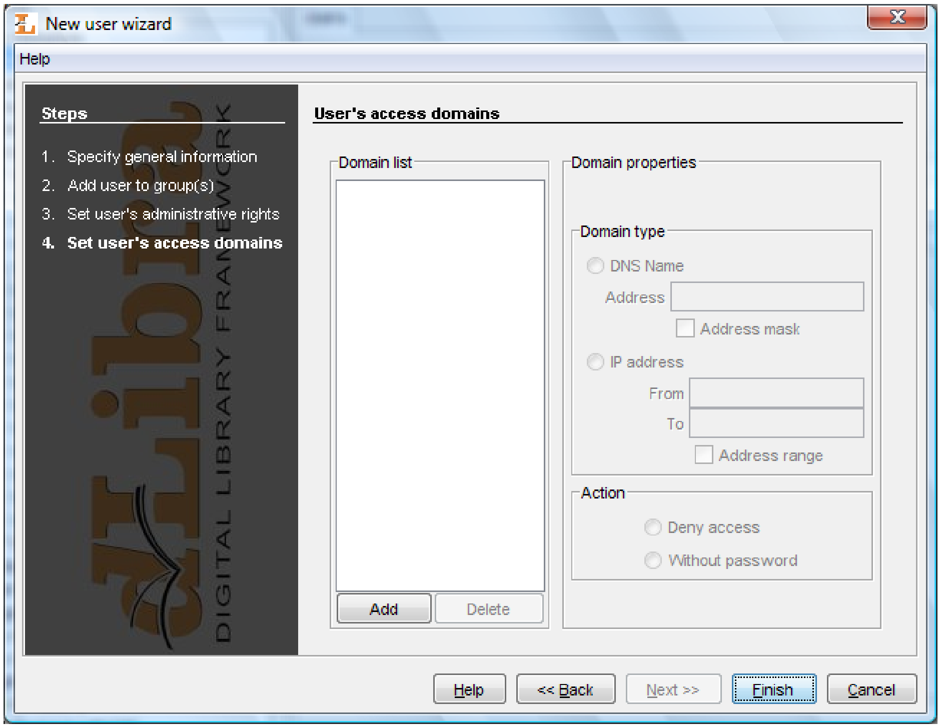
Figure 4.16. New user wizard - assigning administrative rights


New user wizard
Help
Steps
1. Specify general information
2. Add user to group(s)
3. Set user's administrative rights
4. Set user's access domains
dLibra
DIGITAL LIBRARY FRAMEWORK
User's administrative rights
Rights management
Help
<< Back
Next >>
Finish
Cancel

□ Account managing
□ Group managing
□ Attributes managing
□ Attribute values management
□ Languages managing
□ Tags management
□ Collection managing
□ WWW application managing

6. In the last fourth step (Figure 4.17, “New user wizard - defininf access domains”), assign access domains. Managing access domains is described in details in ???.

Figure 4.17. New user wizard - defininf access domains



To remove a user:

1. In the Library Tree Select the user to be removed.

2. Press the 🗑 Delete button. Alternatively, a context menu or a pull-down menu can be used.

4.2.5.2. IP users management

IP users in *dLibra* system are defined by administrators as a set of IP addresses or domain names. Therefore authentication of such a user is performed by checking IP address of a computer that established connection. If the IP address is included in a set of addresses assigned to an existing IP user then *dLibra* system assumes that the connection is established by this IP user. As you can see the authentication process is automatic (user does not have to provide credentials). If there is a need to identify a set of computers (e.g. reading room) as one user then IP user should be used. Then we can assign to such a user (which is a set of computers) rights to publications. For example view rights to specific publications can be granted only to computers in specific location (e.g. specific institution). As a result these publications will not be available to the web users connecting from other locations, but will be available to the web users connecting from this specific institution.

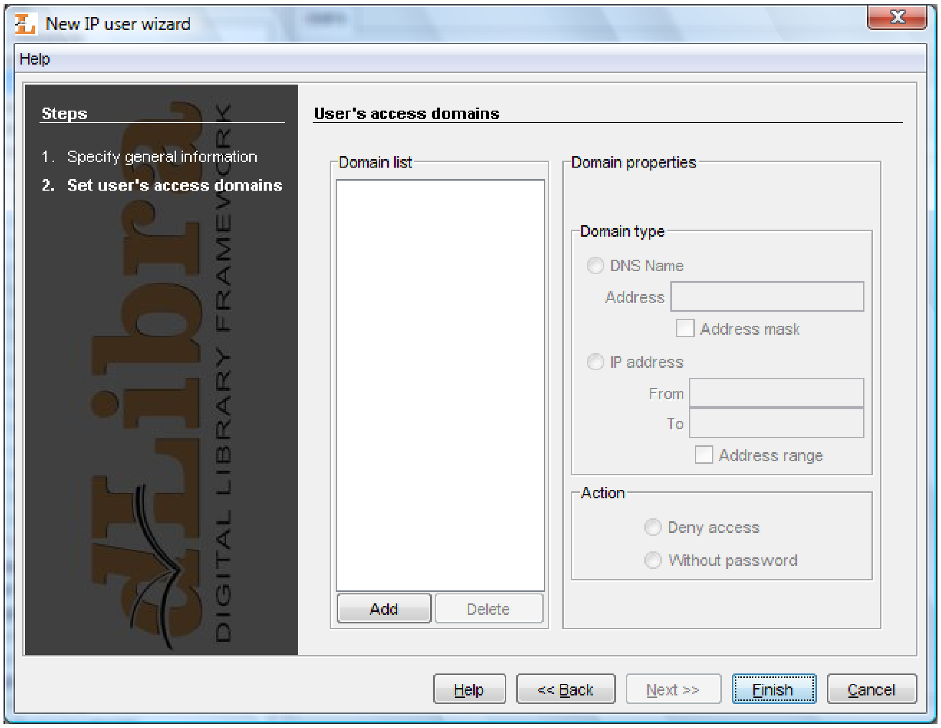
To create a new IP user:

1. In the library tree select the IP users node or any IP user node.

2. From the popup menu select the New IP user... option. Alternativelly choose New user... from Management menu.

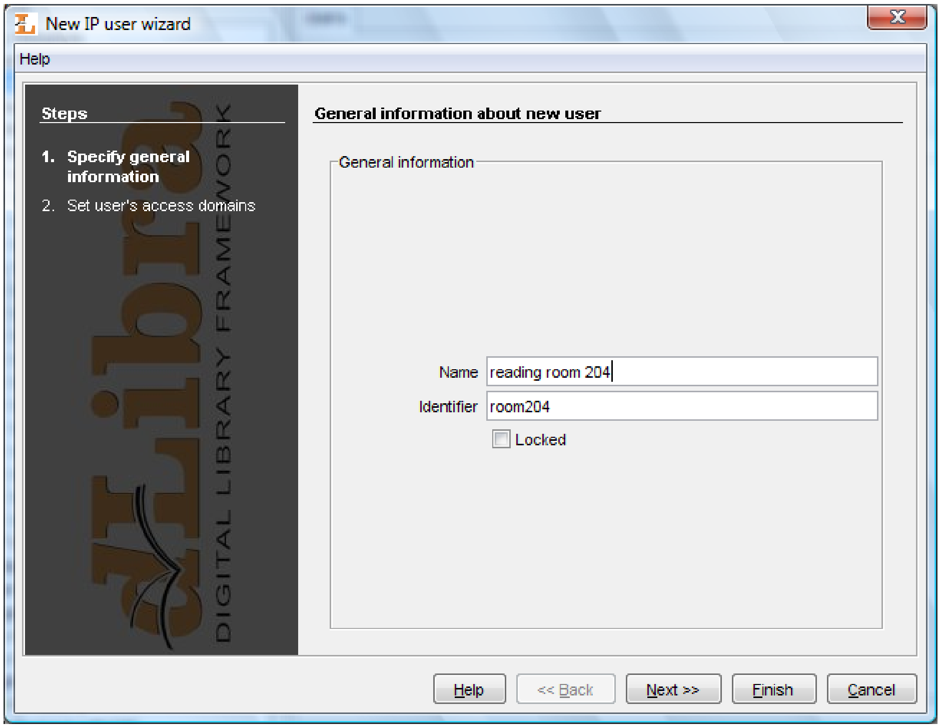
3. In the first step (Figure 4.14, “New user wizard - general information”) provide user's general information. Identifier and name are required. To go to the next step, press the Next button. It is to possible to create the user here by pressing the Finish button.

Figure 4.18. New IP user wizard - general information



4. In the last step (Figure 4.17, “New user wizard - defininf access domains”) assign IP addresses or domains which define the IP user. Managing domains is described in details in ???.

Figure 4.19. New IP user wizard - defininf access domains


New IP user wizard
Help
Steps
1. Specify general information
2. Set user's access domains
dLibra DIGITAL LIBRARY FRAMEWORK
General information about new user
General information
Help
<< Back
Next >>
Finish
Cancel

Name: reading room 204

Identifier: room204

□ Locked

To remove a user:

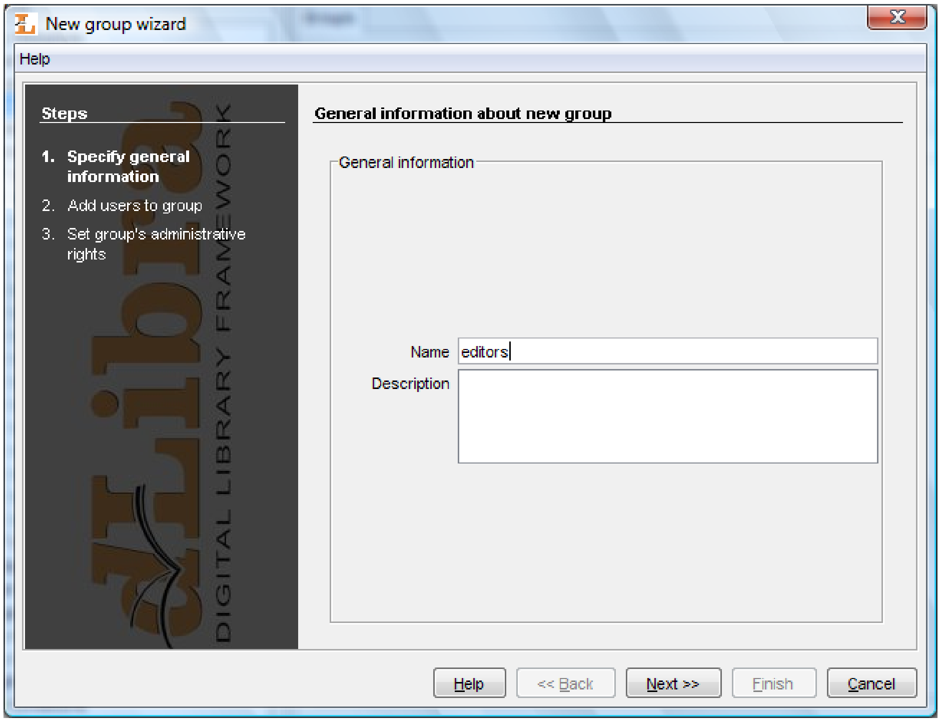
1. In the library tree select the IP user to be removed.
2. Select 🗑 Delete in a context menu or main menu (Edit menu).

4.2.5.3. Library groups management

To create a new users group:

1. In the Library Tree select the groups node or a group node.
2. From the toolbar select ✚ button or choose Nowa grupa.. option from popup menu. You can also choose New group... option from Management menu.
3. In the new group wizard (the first step --- Figure 4.20, “New group wizard - general information”) specify general information (description and name). Press the Next button to go to the next step. It is to possible to create the group at this point by pressing the Finish button.

Figure 4.20. New group wizard - general information

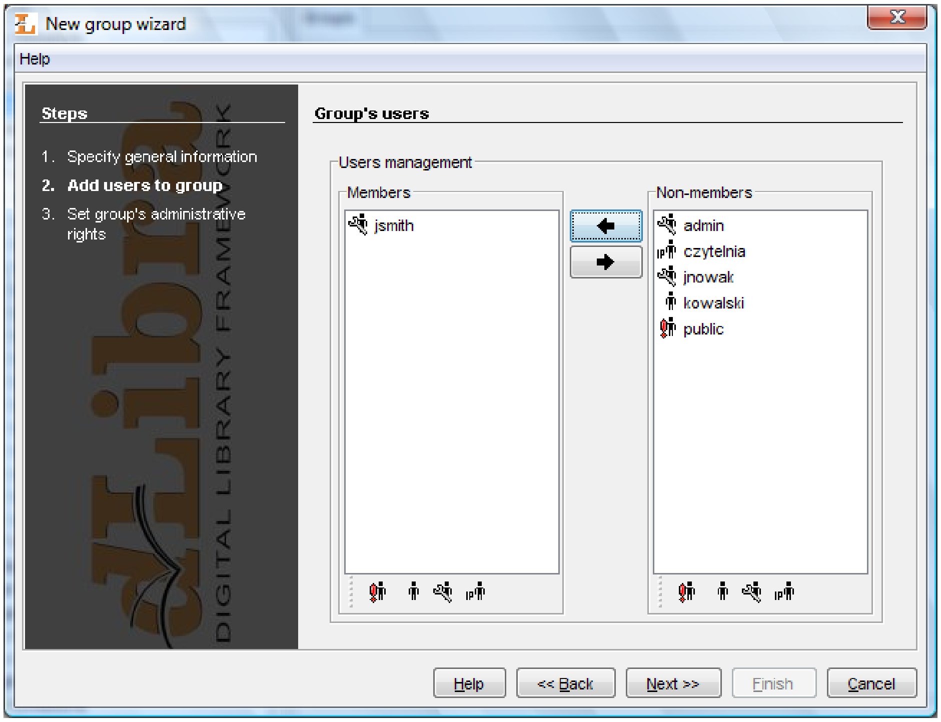

New group wizard
Help
Steps
1. Specify general information
2. Add users to group
3. Set group's administrative rights
DIGITAL LIBRARY FRAMEWORK
General information about new group
General information
Help
<< Back
Next >>
Finish
Cancel

Name: editors

Description:

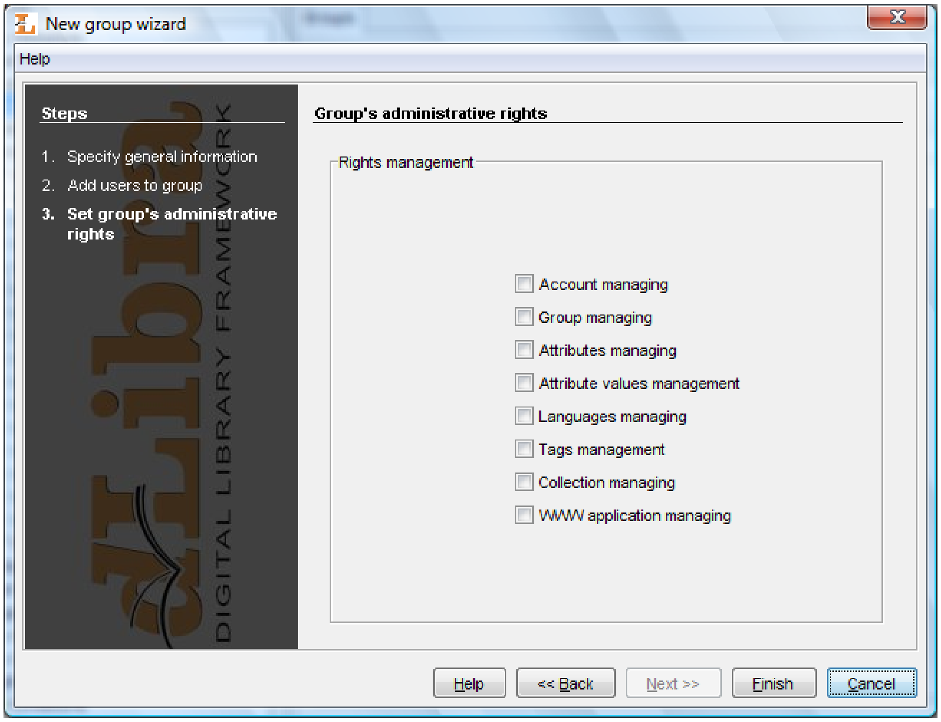
4. In the second step (Figure 4.21, “New group wizard - assigning users”) assign users to group. If a user is placed on the Members list then the user will be assigned to new group. zostaną dodani do nowej grupy. Press the Next button to go to the next step. It is to possible to create the group at this point by pressing the Finish button.

Figure 4.21. New group wizard - assigning users



5. In the last step (Figure 4.22, “New group wizard - assigning administrative rights”) assign administrative rights to group. Detailed information about the meaning of particular right can be found in ???. Press the Finish button to create the group.

Figure 4.22. New group wizard - assigning administrative rights



To remove a group:

1. In the Library Tree select the group to be removed.

2. Press the Delete button. Alternatively, a context menu or a pull-down menu can be used.

4.2.5.4. LDAP groups management

> **Note**
>
> To understand this section basic knowledge about LDAP is required.

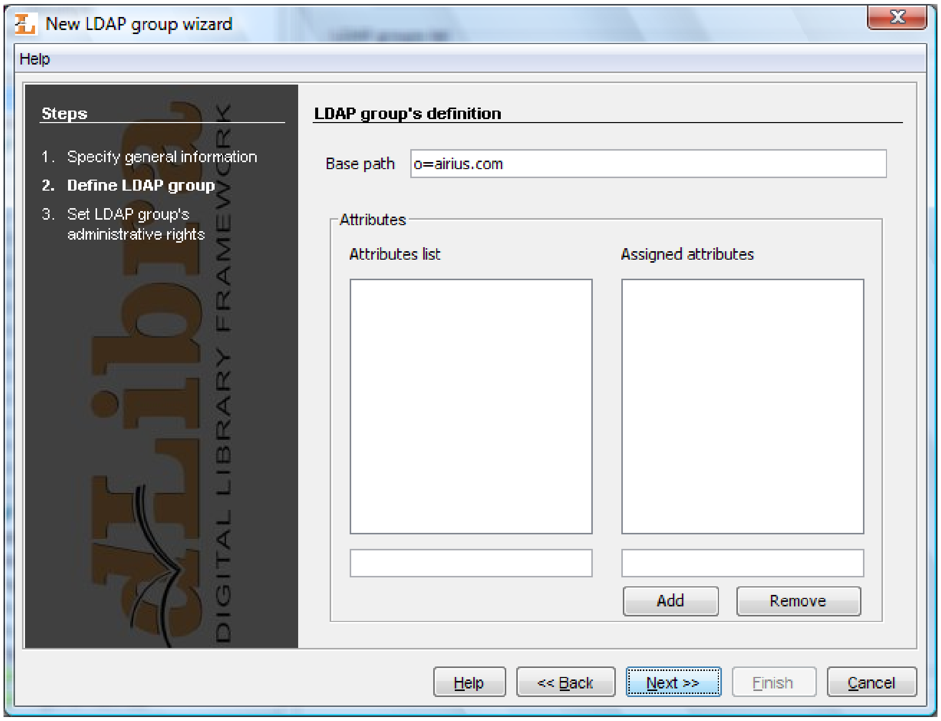
To create a new LDAP group:

1. In the Library Tree select the LDAP groups node or an LDAP group node.

2. Select the button from toolbar or choose New LDAP group... from popup menu. You can also choose New LDAP group... option from Management menu.

3. In the first step of new LDAP group wizard (Figure 4.23, "New LDAP group wizard - general information") specify general information.

Figure 4.23. New LDAP group wizard - general information



New LDAP group wizard

Steps

1. Specify general information
2. **Define LDAP group**
3. Set LDAP group's administrative rights

LDAP group's definition

Base path: o=airius.com

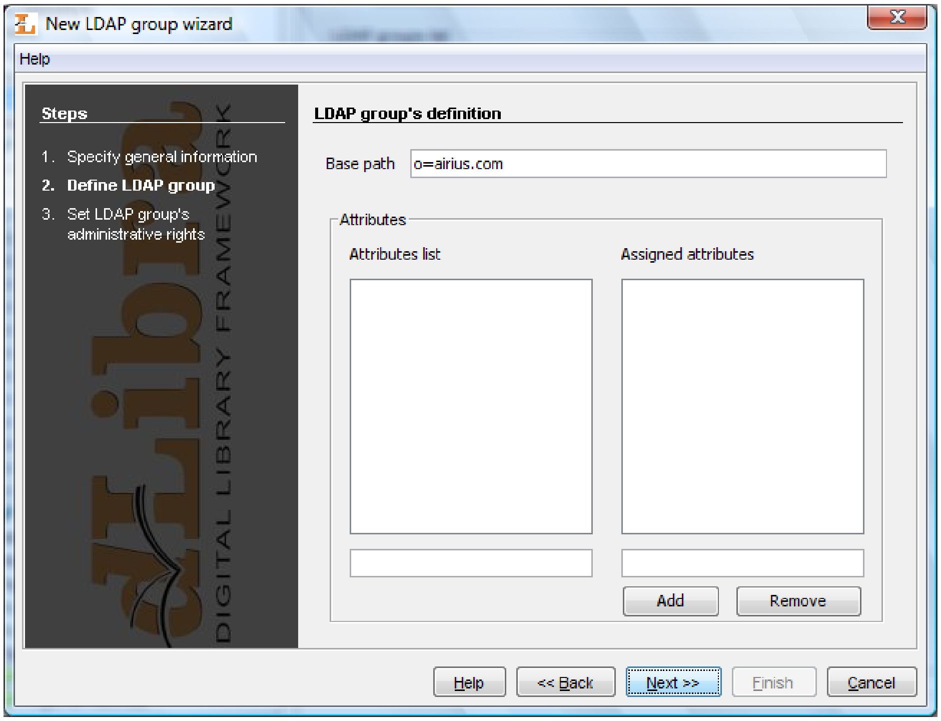
4. In the second step (Figure 4.24, "New LDAP group wizard - defining LDAP group") define group. Assign group attribtues and base path. Attributes are listed in Attributes list. Assigned attribute-value pairs are listed in Assigned attributes list. To add attribute:

a. Enter attribute name (or select it from Attributes list list) in the Attributes list text field,

b. Enter attribute value in the Assigned attributes text field and press Add button.

To remove attribute-value pair, select it on Assigned attributes list and press Remove button. Press the Next button to go to the next step. It is to possible to create the LDAP group at this point by pressing the Finish button.

Figure 4.24. New LDAP group wizard - defining LDAP group



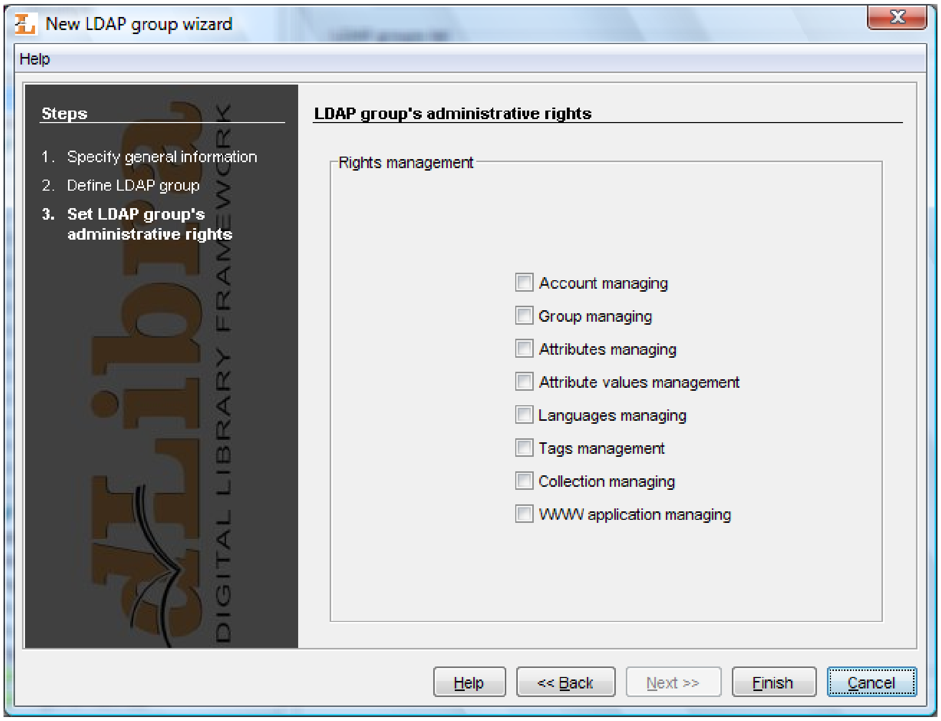
New LDAP group wizard

Help
Steps
1. Specify general information
2. Define LDAP group
3. Set LDAP group's administrative rights
DIGITAL LIBRARY FRAMEWORK
LDAP group's definition

Base path: o=airius.com

5. In the last step (Figure 4.25, “New LDAP group wizard - assigning administrative rights”) assign administrative right. To create new LDAP group press Finish button.

Figure 4.25. New LDAP group wizard - assigning administrative rights



Steps

1. Specify general information
2. Define LDAP group
3. **Set LDAP group's administrative rights**

LDAP group's administrative rights

Rights management

□ Account managing
□ Group managing
□ Attributes managing
□ Attribute values management
□ Languages managing
□ Tags management
□ Collection managing
□ WWW application managing

To remove an LDAP group:

1. In the Library Tree select the LDAP group to be removed.
2. Press the 🗑 Delete button. Alternatively, a context menu or a pull-down menu can be used.

4.2.6. Languages management

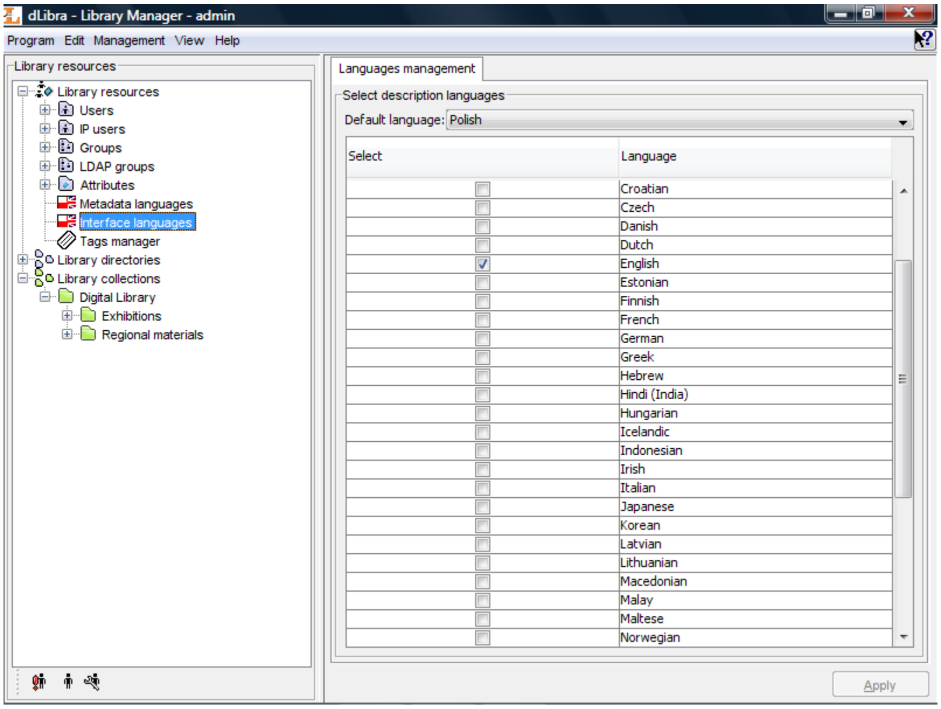
WWW application in *dLibra* system may have different language versions - for example, users visiting digital library from England see WWW pages (buttons' names, labels, etc.) in English while users visitng digital library from Poland see WWW pages in Polish. These languages are interface languages. By default users have two languages avaliable - Polish and English. In order to add new interface language it is required to translate buttons' names, labels, etc. Additionally, administrators have to remember to provide collection names, attributes names, etc. in new interface language (because users see this information on WWW pages).

In *dLibra* system there are also metadata languages. Edition's bibliographic description, edition's description and comment are specified in metadata languages.

On the WWW pages users may choose both interface language and metadata language. The choice of interface language is independent from the choice of metadata language (and vice versa) therefore it is possible that user have WWW pages presented in English but specific edition's metadata in Polish.

When Interface languages or Metadata languages (???) node is selected administrator may set languages that are used in *dLibra* system. Currently used languages are selected on languages table. In order to add/remove language select/deselect check box corresponding to the language. Every language type (metadata and interface) has to have default language. Default language is used when the system cannot find the language corresponding to WWW user. For example, let us assume that we have system with English (default) and Polish interface language and Polish (default) metadata language. When user from Germany visits WWW pages then default (English) interface language and default metadata language (Polish) will be chosen to present data (because there is no German interface/metadata language).

Figure 4.26. Languages management



Select description languages

Default language: Polish

Select	Language
□	Croatian
□	Czech
□	Danish
□	Dutch
☑	English
□	Estonian
□	Finnish
□	French
□	German
□	Greek
□	Hebrew
□	Hindi (India)
□	Hungarian
□	Icelandic
□	Indonesian
□	Irish
□	Italian
□	Japanese
□	Korean
□	Latvian
□	Lithuanian
□	Macedonian
□	Malay
□	Maltese
□	Norwegian

4.3. Advanced right management

4.3.1. Introduction

In the *dLibra* library access restrictions are applied on three different levels: library level, directory level and publication level. The library-level access restrictions concern attribute scheme management, library users and groups management. The directory-level restrictions can be applied to every single directory in the library content tree and regard content visibility, permission to read and edit the content of the directory. Finally, the publication-level access management applies to a single publication and regards publication viewing and reading. The three groups of access permissions will be discussed in detail further in this section.

Regardless of the access level, rights are granted on a user or group basis. A user has the rights he was directly granted, but also the rights inherited from all groups he is a member of. Thus, a change made to access permissions of a group will affect every user belonging to the group.

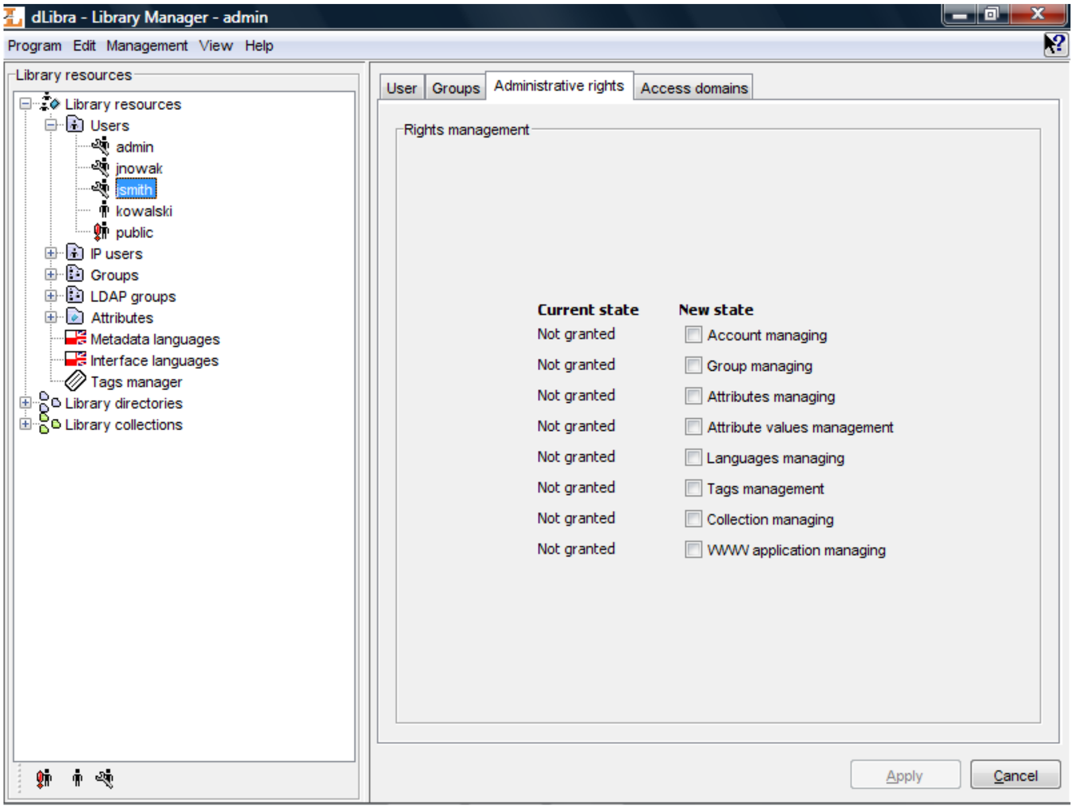
4.3.2. Library-level permissions

In *dLibra* there are five library-level (administrative) permissions:

- account management - allows to create, remove and alter user accounts
- group management - allows to create, remove and alter user groups
- attributes management - allows to create, remove and alter attributes
- attribute values management - allows to manage the synonyms dictionary
- languages management - allows to manage sets of metadata and interface languages
- tags management - allows to manage all tags in administrator application
- collections management - allows to create, remove and alter collections
- web application management - allows to login to administrative panel of web application and use available functions

The administrative rights of a user can be changed in the User Properties panel, Administrative rights tab (Figure 4.27, "User administrative rights tab"). The administrative rights of a group can be changed in the Group Properties panel, Administrative rights tab.

Figure 4.27. User administrative rights tab



Current state	New state
Not granted	□ Account managing
Not granted	□ Group managing
Not granted	□ Attributes managing
Not granted	□ Attribute values management
Not granted	□ Languages managing
Not granted	□ Tags management
Not granted	□ Collection managing
Not granted	□ WWW application managing

4.3.3. Directory-level permissions

In *dLibra*, access to the library directories is controlled separately for every single directory and every single library user. This means that both for the same directory different users can have different permissions, and one user can have different levels of access to different directories. There are four different ways in which a user can have directory permission:

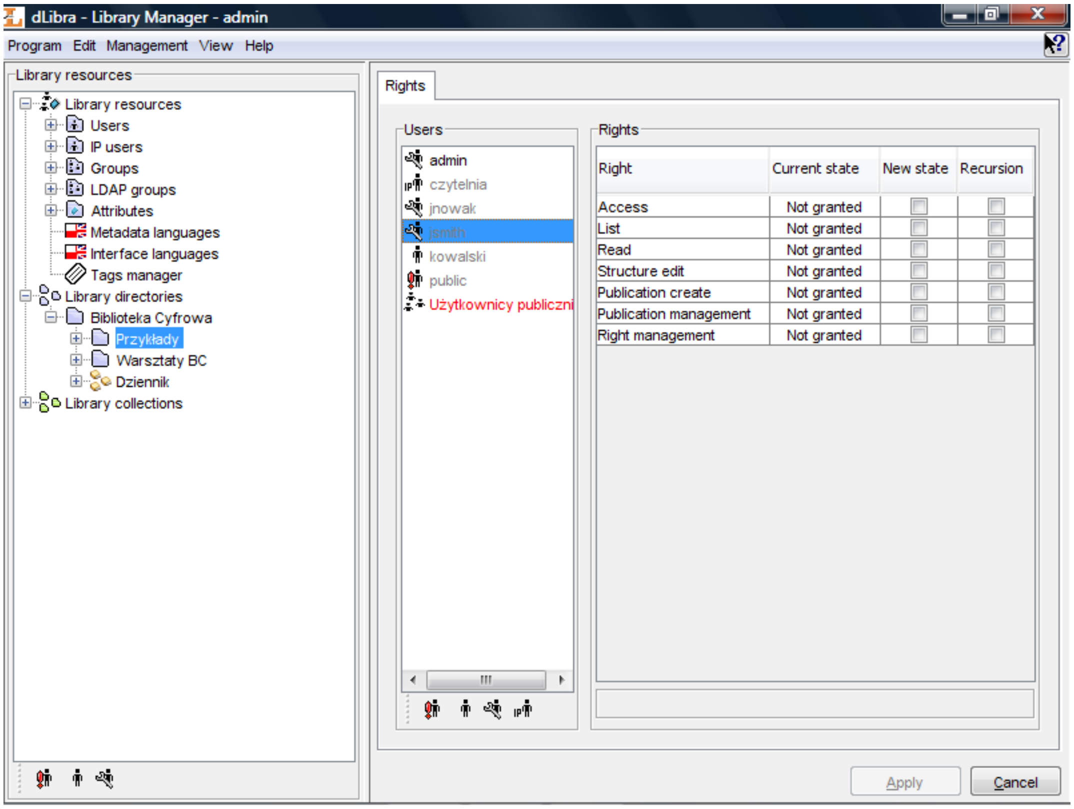
- directly granted
- implied by another right that is directly granted (e.g. the directory listing right is implied by the directory content reading right)
- inherited from one of the parent directories for which the permission is directly granted or implied
- inherited from a group the user is a member of

In *dLibra* there are seven directory-level permissions:

- Access - makes the directory visible in a user's view of the library tree. *Not* inherited from the parent directories.
- List - enables a user to view the content of the directory (i.e. publications and subdirectories) and view the published editions of publications placed in the directory. Inherited from the parent directories. Implies the Access permission.
- Read - enables a user to view all editions (whether published or not) of all publications contained in the directory. Inherited from the parent directories. Implies the Access and List permissions.
- Structure edit - enables a user to edit the structure of the directory, i.e. to create, move and remove subdirectories. Inherited from the parent directories. Implies the Access, List and Read permissions.
- Publication create - enables a user to create new publications in the directory. Inherited from the parent directories. Implies the Access and List permissions.
- Publication management - enables a user to remove publications from the directory. Inherited from the parent directories. Implies the Access, List, Read and Publication create permissions in the current directory and Publication view and manage permissions for all publications in this directory.
- Rights management - enables a user to alter access rights for the directory. Inherited from the parent directories. Implies the Access, List and Read permissions.

The directory-level access permissions can be altered in the Directory Properties panel, Rights tab.

Figure 4.28. Directory-level rights editor



Library resources
- Library resources
  - Users
  - IP users
  - Groups
  - LDAP groups
  - Attributes
  - Metadata languages
  - Interface languages
  - Tags manager
- Library directories
  - Biblioteka Cyfrowa
    - Przykłady
    - Warsztaty BC
    - Dziennik
- Library collections

Users
- admin
- czytelnia
- jnowak
- jsmith
- kowalski
- public
- Użytkownicy publiczni

Right	Current state	New state	Recursion
Access	Not granted	□	□
List	Not granted	□	□
Read	Not granted	□	□
Structure edit	Not granted	□	□
Publication create	Not granted	□	□
Publication management	Not granted	□	□
Right management	Not granted	□	□

On the list on the left library users na (in black having at least one right, in grey having no rights) and groups (in red) are shown. When the 🧍 button is pressed, public users are not displayed. After selecting one or more users or groups, their rights to the chosen directory are displayed, in four columns, in the table on the right:

- Right - contains the name of the permission
- Old state - displays the state of the permission before change
- New state - displays the state of the permission after change. To grant the right to the chosen users - select the checkbox in the table.
- Recursion - select the checkbox to grant the right directly for all subdirectories of the selected directory.

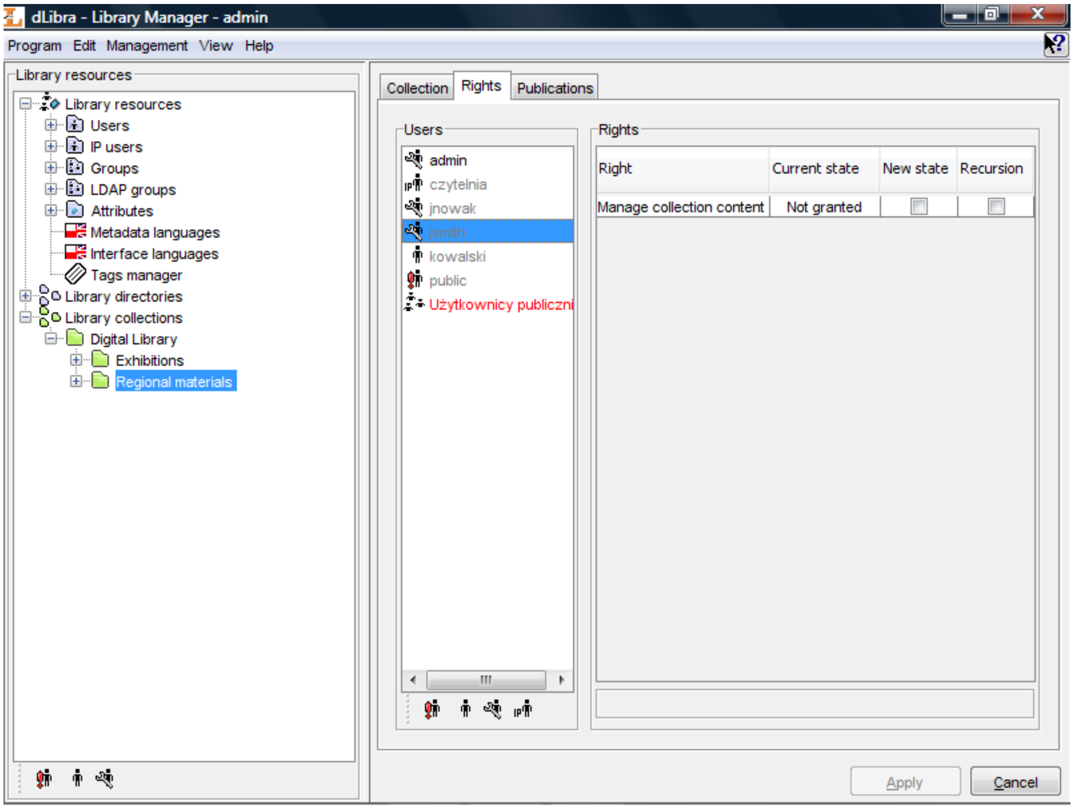
4.3.4. Collection-level permissions

As with the library directories, access to *dLibra* publications is controlled separately for every single publication and every single library user. Similarly to the library directories, access permissions to publications can be directly granted, implied, inherited from parent directories or inherited from a group (see previous section).

In *dLibra* there is only one publication-level permission:

- Manage collection content - enables a user to add and remove publications from a chosen collection.

Figure 4.29. Collection-level rights editor



Right	Current state	New state	Recursion
Manage collection content	Not granted	□	□

Rights can be managed on a tab signed Rights.

4.3.5. Publication-level permissions

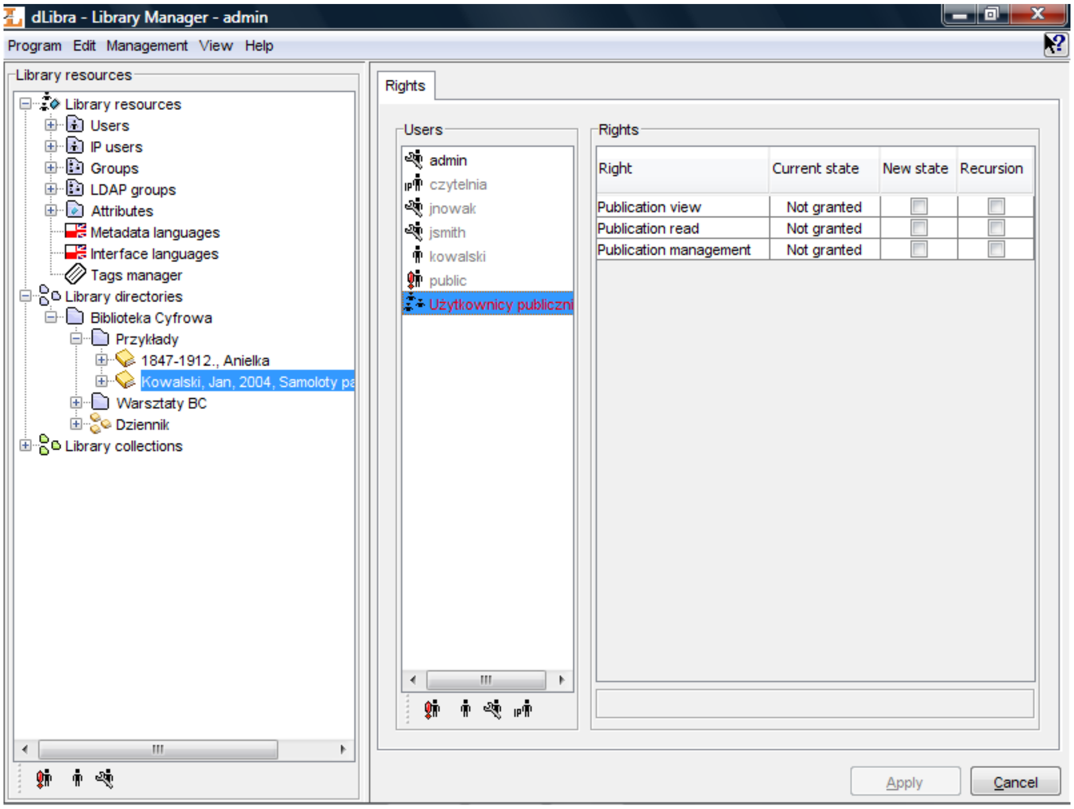
As with the library directories or collections, access to *dLibra* publications is controlled separately for every single publication and every single library user - for the same publication different users can have different permissions, and one user can have different levels of access to different publications. Similarly to the library directories, access permissions to publications can be directly granted, implied, inherited from parent directories or inherited from a group (see previous section).

In *dLibra* there are three publication-level permissions:

- View - enables a user to read the published editions of a publication. Inherited from the "List" right of the directory the publication belongs to.
- Read - enables a user to read all editions (whether published or not) of a publication. Inherited from the "Read" right of the directory the publication belongs to.
- Manage - enables a user to alter other user's rights for it. By default granted to the creator of the publication.

The publication-level access permissions can be altered in the Publication Properties panel, Rights tab.

Figure 4.30. Publication-level rights editor



Right	Current state	New state	Recursion
Publication view	Not granted	☐	☐
Publication read	Not granted	☐	☐
Publication management	Not granted	☐	☐

4.4. Access domains management

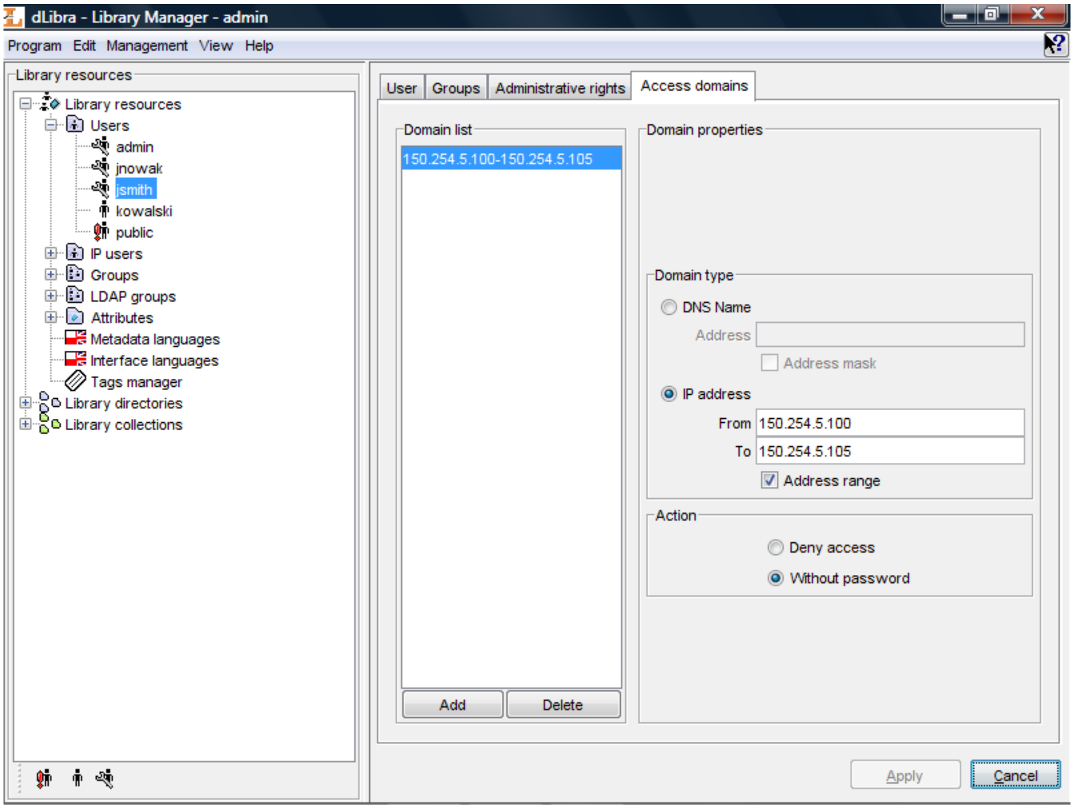
In *dLibra*, for every library user it is possible to define a number of access domains so that, from some places in the Internet, the user can access the library without authentication, from other - access is denied or requires entering a password.

The most important part of an access domain definition is the set of addresses the access restrictions should apply to. The address can be defined by specifying:

- a single DNS name (e.g. rose.man.poznan.pl)
- a DNS address mask (e.g. *.poznan.pl)
- a single IP address (e.g. 62.21.18.55)
- a range of IP addresses (e.g. 62.21.17.55 - 62.21.18.55)

For every library user any number of rules: address -> action can be defined. The action can be either to allow access without authentication (without entering a password) or to deny access. The rules can be defined in User Properties panel, Access domains tab.

Figure 4.31. User access domains



Domain list: 150.254.5.100-150.254.5.105

Domain type:

○ DNS Name

Address:

☐ Address mask

● IP address

From: 150.254.5.100

To: 150.254.5.105

☑ Address range

Action:

○ Deny access

● Without password

Use the Add button to add new access domains and the Delete to remove access domains. In the Domain Properties panel - address type and action can be specified.

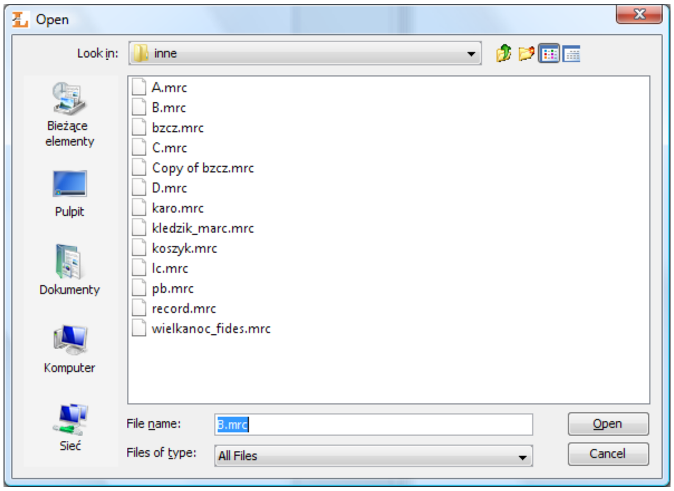
Appendix A. Importing groups of values from MARC files

Groups import configuration is entirely based on metadata import from MARC files. The difference is that in case of this extension instead of *dLibra* attributes names the following names have to be provided:

- baseValue - value imported to this element will state the group name which is to be imported. If there is more than one value available then first value will be used. Therefore this configuration should import single value.
- alternativeValues - values imported to this element will be the synonyms in the group imported in *baseValue*.

If the editor uses this extension to import groups of values a file/directory select dialog opens (Figure A.1, “Dictionary import - select directory/files”). If the editor selects a directory then all files in the directory are processed to extract groups of values. If the editor selects specific files then these files are processed.

Figure A.1. Dictionary import - select directory/files



Look in: inne

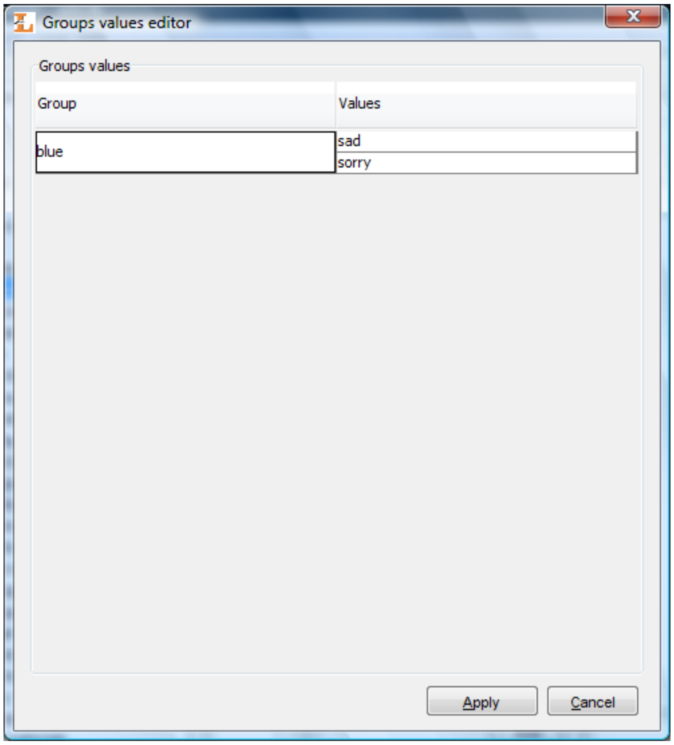
A.mrc
B.mrc
bzcz.mrc
C.mrc
Copy of bzcz.mrc
D.mrc
karo.mrc
kledzik_marc.mrc
koszyk.mrc
lc.mrc
pb.mrc
record.mrc
wielkanoc_fides.mrc

File name: B.mrc

Files of type: All Files

After processing selected diretory or files the application opens a dialog where a table of all retrieved groups of values is displayed (Figure A.2, “Dictionary import - groups of values”). Base value of a group is placed in the Group column, values which are in the group are placed in the same row in Values column. Base values and regular values may be edited.

Figure A.2. Dictionary import - groups of values



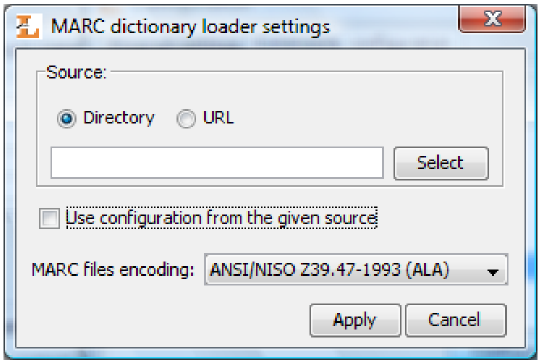
Groups values editor

Groups values

Group	Values
blue	sad sorry

In order to use new configuration files open extension's configuration dialog (Figure C.1, “Metadata import from MARC files - configuration”, steps to open dialog are described in configuration section). When the configuration dialog is open, select the source of new configuration files (file on computer drive or URL) and check Use configuration from given source check box. Additionally, it is required to specify encoding of MARC files which will be imported. MARC file encoding depends on the information system it comes from.

Figure A.3. Dictionary import - configuration dialog



MARC dictionary loader settings

Source:

● Directory ○ URL

□ Use configuration from the given source

MARC files encoding: ANSI/NISO Z39.47-1993 (ALA)

Appendix B. Retrieving metadata through the Z39.50 extension

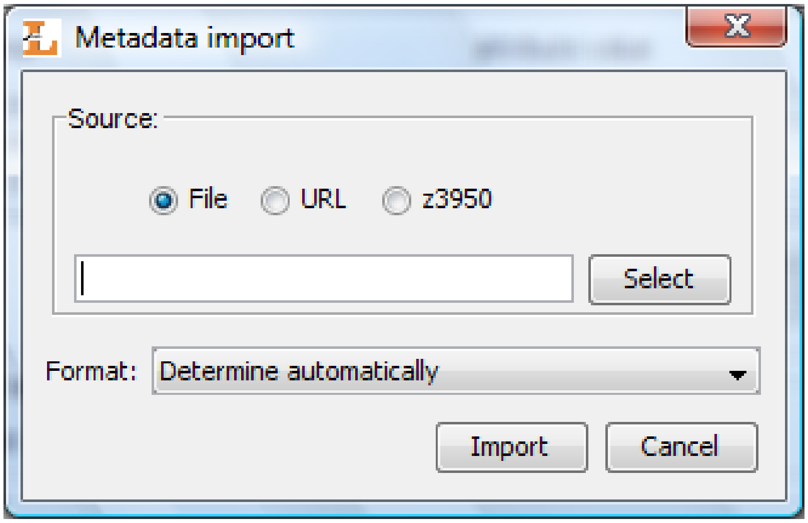
> **Important**
>
> For better understanding of the metadata retrieving mechanism through the Z39.50 protocol, please read section describing metadata import mechanis which is available in bibliographic description editor. Moreover the reader should be familiar with the Z39.50 protocol.

Many library systems provides metadata through the Z39.50 protocol. Z39.50 extension allows editors to retrieve metadata from the systems which provide the metadata through the Z39.50 protocol. Z39.50 protocol is a complex standard which supports transfering metadata in different formats. Z39.50 extension supports retrieval of metadata only in MARC 21 communication format (definitely it is *usmarc*). Metadata retrieved using Z39.50 extension may be then imported using MARC import extension.

In order to retrieve metadata using Z39.50 extension:

1. In metadata editor on the Import/Export metadata panel press Import... button, which will cause metadata import dialog (Figure B.1, “Metadata import dialog”) to open. On the metadata import dialog select Z39.50 option. This option determines the usage of the Z39.50 extension in metadata file selection process.

Figure B.1. Metadata import dialog



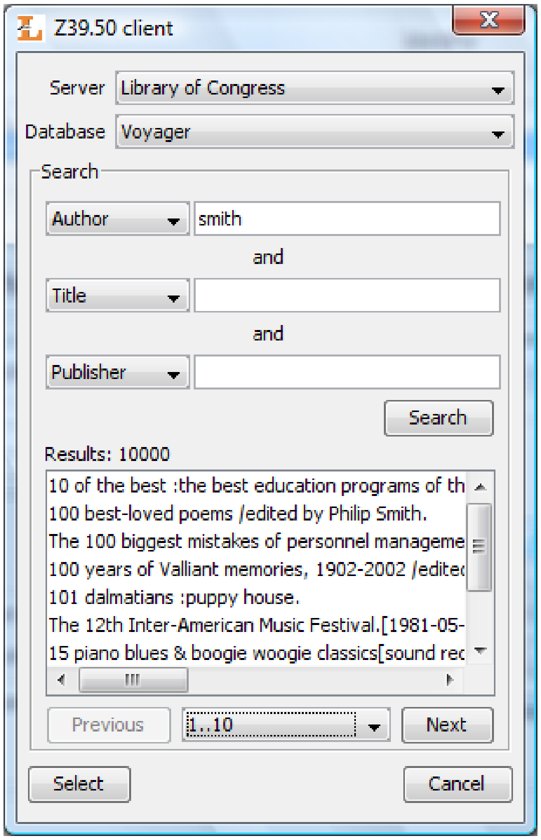
Source: ● File ○ URL ○ z3950

Format: Determine automatically

2. In order to select metadata to import press Select... button. Metadata search dialog (Figure B.2, “Z39.50 metadata search dialog”) will open. On this dialog editor may choose Z39.50 server and database which will be searched. Search query is build based on three values/phrases which are typed by the editor in the text fields placed on the Search panel. The attributes which will be used to search may be chosen by the editor using the combo box near search text field. On the presented metadata search dialog (Figure B.2, “Z39.50 metadata search dialog”) chosed attributes which will be used to search are author, title and publisher. In order to start search process press Search button. When the search process is finished it is possible to browse

search results using Previous and Next buttons which respectively move to the previous or next page of search results. Editor may also directly select specific search results page using the combo box placed between Previous and Next button.

Figure B.2. Z39.50 metadata search dialog



Z39.50 client

Server: Library of Congress

Database: Voyager

Search

Author: smith

and

Title:

and

Publisher:

Results: 10000

10 of the best :the best education programs of th
100 best-loved poems /edited by Philip Smith.
The 100 biggest mistakes of personnel manageme
100 years of Valliant memories, 1902-2002 /editec
101 dalmatians :puppy house.
The 12th Inter-American Music Festival.[1981-05-
15 piano blues & boogie woogie classics[sound rec

1..10

3. In order to retrieve metadata file select specific position on the search results list and press Select button. The metadata file will be retrieved and saved on the local drive, Z39.50 metadata search dialog will be closed and the metadata import dialog will be automatically filled with a path to retrieved metadata file. Next steps should be performed according to the information enclosed in the metadata import section (the simplest case is simply pressing the Import button on the metadata import dialog).

Z39.50 extension may be configured to meet the needs of a specific installation. Configuration file for the Z39.50 extension is named `z3950_servers.xml`.This file defines i.a. Z39.50 servers, databases, attributes and logical operator which will be used to search for metadata. Default Z39.50 configuration file is presented below.

```
<?xml version="1.0" encoding="UTF-8"?>
<servers>
 <server>
  <name>Library of Congress</name>
  <name lang="en">Library of Congress</name>
  <name lang="pl">Biblioteka Kongresu Stanów Zjednoczonych</name>
  <host>z3950.loc.gov</host>
  <port>7090</port>
  <databases>
```

```
   <database>Voyager</database>
  </databases>
  <recordencoding>MARC-8</recordencoding>
  <queries>
   <attrset>@attrset bib-1</attrset>
   <operator>@and</operator>
   <query>
    <name>Author</name>
    <name lang="pl">Autor</name>
    <name lang="en">Author</name>
    <searchquery>@attr 1=1003 $1</searchquery>
   </query>
   <query>
    <name>Title</name>
    <name lang="pl">Tytuł</name>
    <name lang="en">Title</name>
    <searchquery>@attr 1=4 $1</searchquery>
   </query>
   <query>
    <name>Publisher</name>
    <name lang="pl">Wydawca</name>
    <name lang="en">Publisher</name>
    <searchquery>@attr 1=1018 $1</searchquery>
   </query>
   <query>
    <name>Everywhere</name>
    <name lang="pl">Wszędzie</name>
    <name lang="en">Everywhere</name>
    <searchquery>@attr 1=1035 $1</searchquery>
   </query>
  </queries>
 </server>
</servers>
```

The file is in an XML format. The main node in a file is `<servers>` where all Z39.50 servers definitions are placed (the number of servers is not restricted). Each server is defined in the scope of `<server>` tag where the following tags should be provided:

- `<name>` - this is the name of the server presented on the Z39.50 metadata search dialog. The name can be specified for different languages by using the `lang` attribute in `<name>` tag. The value of the `lang` attribute should be a two letter symbol of the language (according to the ISO 639 standard) in which the name of the server is specified. Default server name is specified in the `<name>` tag which does not have the `lang` attribute. Please notice that particular names are defined in separate `<name>` tags.

- `<host>` - this is the Internet address of the Z39.50 server (IP address or domain name of the server).

- `<port>` - this is the port of the Z39.50 server.

- `<databases>` - this tag encloses the list of `<database>` tags which are the names of the databases that can be used to search for metadata.

- `<recordencoding>` - this is the encoding of metadata records which are provided by the Z39.50 server.

- `<queries>` - this is the node where all the information concerning search panel are placed. All the values which start with the @ character are connected with query languages used in JZKit 2 library (http://jzkit.org/). And so:

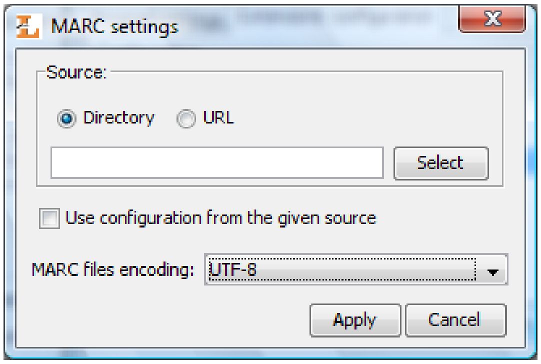
- `<attrset>` - defines the Z39.50 set of attributes used to search.
- `<operator>` - defines the logical operator used to create a query by combinig the values provided in search fields using the operator.
- `<query>` - specifies the name of the attribute which will be used to search and the query which is connected with this attribute. The name of the attribute may be provided in many languages using the `lang` attribute (in the same way it is for the server name). Default name of the attribute is enclosed in the `<name>` tag which does not have `lang` attribute. The search query is specified in the `<searchquery>` tag, where the $1 element is replaced with the value from corresponding search field.

In order to change configuration of the Z39.50 extension, please follow the instructions provided in the application configuration section.

Appendix C. MARC metadata import

By default, attributes' values import from MARC 21 communication format is based on built-in configuration. It is also possible to use external configuration (defined in text files). The text files are simply property files. In order to use new configuration files open extension's configuration dialog (Figure C.1, “Metadata import from MARC files - configuration”, steps to open dialog are described in configuration section). When the configuration dialog is open, select the source of new configuration files (file on computer drive or URL) and check Use configuration from given source check box. Additionally, it is required to specify encoding of MARC files which will be imported. MARC file encoding depends on the information system it comes from.

Figure C.1. Metadata import from MARC files - configuration



MARC settings

Source:

● Directory ○ URL

□ Use configuration from the given source

MARC files encoding: UTF-8

C.1. `marcImport.properties` configuration file format

```
Title=245:${a} ${b} ${n};130;210;222;240;246;730;740;
Creator=100;110;111;
Subject=
Description=6XX;
Publisher=260a;260b;260f;
Contributor=700;710;711;
Date=260c;
Type=
Identifier=920;856u;
Source=
Language=041;546;008/35-37
Relation=250;534;440;490;800;810;811;830;
Coverage=
Rights=506;540;
```

Examplary configuration file is presented above. This file contains configuration which defines assignment of MARC elements to *dLibra* attributes.

Every line in the configuration file contains configuration for single attribute. Every line is composed of RDF attribute name, the equal sign and a list of MARC elements to assign to the attribute. If the RDF name is preceeded by the language name and dot (e.g. `en.Title=210;222;240;246;730;740;`) then the following after RDF name rule will be used to import metadata to this language. If there is no language name then the rule is used to import metadata to user selected language (selected language tab in metadata editor). The name of the language has to be two-letter code according to ISO 639 standard. RDF name of an attribute can be found in administrator application (editing panel for an attribute). MARC elements which may be imported are, inter alia, subfield value, characters from control fields etc. If attribute's RDF name is not specified no values will be assigned to the attribtue.

A record defining a MARC field number which value is to be imported into attribute value has the following basic syntax: **`AAAb;`**, where **`AAA`** is a three-digit number and **`b`** is a subfield code. It is also possible to combine MARC subfield or extract a range of characters from control fields. Note that the sign **`;`** (semicolon) is a part of this syntax and is necessary for a proper configuration.

It is possible to omit the subfield code as well as use the multi-value code. Details and examples of it are presented below.

- **`100;`** - an example of field number.

  Such a record will either import the value of a the field (note that some fields in MARC format, for instance control fields which numbers are smaller than 010, never have subfields) or import the values of all subfields of this field into an attribute value. Every subfield value will be imported as a separate attribute value.

- **`260c;`** - an example of filed number with subfield code.

  Such a record will import just the value of a certain subfield into attribute value.

- **`6XX;`** - an example of multi-value code.

  Such a record will import the values of all fields and subfields at range 600 - 699. In this way you cannot specify certain subfield codes. It is also possible to define for instance such a record: **`65X;`** , which will analogically import values from fields at range 650 - 699.

- **`245:${a} ${b} ${n};`** - an example for combining MARC subfields into one value.

  We can split this entry into two parts which are separated by the “:” (colon) character:

  1. **`245`** - field number which subfields will be combined into a value

  2. **`${a} ${b} ${n}`** - template which defines how to combine the subfields.

  The entry **`${a}`** means that in its place value from “a” subfield should be placed. The subfield is a subfield of field number placed before the “:” character - in this case it is **`245`** field. So the **`245:${a} ${b} ${n};`** template will combine **`245`** field's subfields (a, b and n) in one value. These subfields will be separated with space (as specified in the template). For example if the subfield 245a has “first value” value, subfield 245b has “second value” value and subfield 245n has “third value” value then the result will be “first value second value third value”. If there is a need to separate these values with anothed character (not space) place them instead of the space in the template (e.g. **`245:${a}-${b} subfield n: ${n};`**). There are few exceptions --- characters “;” (semicolon), “\” (backslash) and “$” - to interpret these characters correctly by the application two additional backslashes have to be placed before (e.g. **`245:${a} ${b}\\;${n};`**).

- **008/35-37** - concerns only control fields - it means extracting a range of chararcers from the control field.

  This template is combined from two parts separated by the slash character (“/”):

  1. **008** - the number of control field which range of characters will be extracted from

  2. **35-37** - this is the rance of characters which will be extracted from the filed number which is placed before slash (“/”).

  The entry means that the character on the position 35, 36 and 37 from the 008 control field will make the value. If the 008 control field, on the 35th position has an “e” character, on the 36th position has an “n” character and on the 37th position has a “g” character then the value of such a entry will be “eng”. If it is needed to extract only one character from a given position simply specify the character position after the slash character, e.g. **008/30**.

C.2. marcImpRemChars.properties configuration file format

```
end-245b=a|b
begin-245a=OS/2
end-260c=c
```

An examplary marcImpRemChars.properties file is presented above.

This file enables user to define characters (or series of characters) to remove from values of MARC subfield before this value is imported to bibliographic description. The characters can be removed both from begin and end of the subfield. Characters are defined using regular expression. Regular expressions which may be used in this mechanism have to be conformant with Java∢ regular expressions (details can be found here).

Line end-245b=a|b means that from the end (the word end) of the 245b subfield “a” or “b” character will be removed (if of course one of these characters will be found at the end of the 245b subfield). Minus separates the place (begin or end) from which we remove the characters and the subfield (245b) from which we remove the characters. After the equals character regular expression which defines characters to remove is specified.

Let us analyze the following example: “begin-245a=ab”. This record will couse that the application will remove the ab characters sequence from the beggining of 245b subfield, if this sequence will be found at the beggining of this subfield. So if in the MARC file subfield 245a will have “abStories” value then the application will transform it to “Stories” value which will be imported to the bibliographic description.

C.3. Default configuration

By default, MARC extension is configured using the following files:

- marcImport.properties:

  ```
  Title=245;130;210;222;240;246;730;740;
  Creator=100;110;111;
  ```

```
Subject=
Description=6XX;
Publisher=260a;260b;260f;
Contributor=700;710;711;
Date=260c;
Type=
Identifier=920;856u;
Source=
Language=041;546;
Relation=250;534;440;490;800;810;811;830;
Coverage=
Rights=506;540;
```

- `marcImpRemChars.properties`

  This file is empty by default.

Appendix D. Export to RDF format

> **Note**
>
> This appendix describes configuration files used for exporting bibliographic description to RDF format. It is dedicated to *dLibra* administrators.

RDF - Resource Description Framework is a format for specifying properties of a given subject. RDF format is used in *dLibra* to specify metadata (bibliographic description). In this case the subject is an element (e.g. publication) and the properties are specified as attributes' values. More about RDF format can be found here.

D.1. RDF export configuration file

There is one *RDF format* configuration file:

- `systemurl.properties` - this file should contains an URL which specified the namespace for exported attribtue values

This file is a properties file, so in each line there is a key and its value separated by the equals character (=), e.g.

```
property=value
```

Configuration file should contain one line which specifies an URL defining the namespace for attribute values. The key which identifies this URL is *systemURL*, for example:

```
systemURL=http://dlibra.psnc.pl/
```

The URL is usually WWW address of specific digital library. Exported file may look as follows:

```
<?xml version="1.0"?>
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:dlibra_avs="http://dlibra.psnc.pl/">
  <rdf:Description rdf:about="http://dlibra.psnc.pl/publication/376">
    <dlibra_avs:Title xml:lang="pl">Przykładowa publikacja</dlibra_avs:Title>
    <dlibra_avs:Author xml:lang="pl">Jan Kowalski</dlibra_avs:Author>
    <dlibra_avs:Format xml:lang="en">text/plain</dlibra_avs:Format>
    <dlibra_avs:Format xml:lang="pl">text/plain</dlibra_avs:Format>
  </rdf:Description>
</rdf:RDF>
```

Default `systemurl.properties` file contains the following entry:

```
systemURL=http://www.dlibra.psnc.pl
```

Appendix E. XML metadata import

Editor/administrator application functionality may be extended with so called extensions. It is possible to add extension which imports metadata from external sources. XML import extension is an extension of this type. It allows importing bibliographic description from XML files (details about XML can be found here).

XML metadata import extension uses XQuery language (details about XQuery can be found here).

To make the import avaliable, appropriate configuration for the XML extension has to be provided. By default, XML extension configuration allows editor to import metadata from RDF and MASTER format (both formats use XML to present metadata).

XML extension is configured using two property files (property files contain key=value pairs):

- `tests.properties` - file contains XQuery queries which test what kind of file is to be imported,
- `conversion.properties` - file contains rules for conversion XML metadata files to *dLibra* metadata.

Relation between both files is very strict - for every XQuery test in `tests.properties` file there are corresponding conversion rules in `conversion.properties` file. For a given metadata XML file, import mechanism performs each test query from `tests.properties` file. If the result have one or more values then metadata are imported from XML file using conversion rules (corresponding to successful XQuery test) from `conversion.properties` files.

`tests.properties` file contains XQuery queries which test whether the metadata file may be imported using corresponding to XQuery test conversion rules. The key identifies conversion rules in `conversion.properties` file.

For example, let us assume that we have the following files (this example presents extension's default configuration):

`tests.properties` file:

```
master=for $x in fn:doc({document})/*[fn:compare(fn:name(), 'msDescription')=0] return $x
rdf_dc=for $x in fn:doc({document})/*[fn:compare(fn:local-name(), 'RDF')=0] return $x
```

`conversion.properties` file:

```
master.Title=for $x in fn:doc({document})//msHeading/title return $x
master.Creator=for $x in fn:doc({document})//msHeading/author return $x
master.Description=for $x in fn:doc({document})//msContents/overview return $x
master.Publisher=for $x in fn:doc({document})//msContents/respStmt/resp/name return $x
master.Contributor=for $x in fn:doc({document})/msDescription/msContents/respStmt//resp return $x
master.Date=for $x in fn:doc({document})//msHeading/origDate return $x
master.Type=for $x in fn:doc({document})//physDesc/form return $x
master.Identifier=for $x in fn:doc({document})//msIdentifier/country/settlement/repository/idno return
master.Source=for $x in fn:doc({document})//msPart//idno return $x
master.Language=for $x in fn:doc({document})//msContents/textLang return $x
master.Language=for $x in fn:doc({document})//msContents/textLang/@otherLangs return $x
master.Rights=for $x in fn:doc({document})//msIdentifier/repository return $x
```

```
rdf_dc.Title=for $x in fn:doc({document})//*[fn:local-name()='Description']/*[fn:local-name()='Title']
rdf_dc.Creator=for $x in fn:doc({document})//*[fn:local-name()='Description']/*[fn:local-name()='Creato
rdf_dc.Subject=for $x in fn:doc({document})//*[fn:local-name()='Description']/*[fn:local-name()='Subjec
rdf_dc.Description=for $x in fn:doc({document})//*[fn:local-name()='Description']/*[fn:local-name()='De
rdf_dc.Publisher=for $x in fn:doc({document})//*[fn:local-name()='Description']/*[fn:local-name()='Publ
rdf_dc.Contributor=for $x in fn:doc({document})//*[fn:local-name()='Description']/*[fn:local-name()='Co
rdf_dc.Date=for $x in fn:doc({document})//*[fn:local-name()='Description']/*[fn:local-name()='Date'] re
rdf_dc.Type=for $x in fn:doc({document})//*[fn:local-name()='Description']/*[fn:local-name()='Type'] re
rdf_dc.Identifier=for $x in fn:doc({document})//*[fn:local-name()='Description']/*[fn:local-name()='Ide
rdf_dc.Source=for $x in fn:doc({document})//*[fn:local-name()='Description']/*[fn:local-name()='Source'
rdf_dc.Language=for $x in fn:doc({document})//*[fn:local-name()='Description']/*[fn:local-name()='Langu
rdf_dc.Relation=for $x in fn:doc({document})//*[fn:local-name()='Description']/*[fn:local-name()='Relat
rdf_dc.Coverage=for $x in fn:doc({document})//*[fn:local-name()='Description']/*[fn:local-name()='Cover
rdf_dc.Rights=for $x in fn:doc({document})//*[fn:local-name()='Description']/*[fn:local-name()='Rights'
```

As we can see, `conversion.properties` file contains conversion rules which correspond to tests in `tests.properties` file. The key in `conversion.properties` file is composed of the key from `tests.properties` file, dot and *dLibra* attribtue's identifier (RDF name). Values resulted from queries which are in `conversion.properties` file will be assigned to attributes with specific RDF name.

Let us assume that we want to import file A which contains metadata in XML format. Import mechanism performs XQuery queries which are placed in `tests.properties` file. The first test which results with non-empty list of values decides which conversion rules will be applied for metadata import. Let us assume that it was a test which key is `metadata`. Import mechanism chooses conversion rules from `conversion.properties` file - all keys which start with `master`. Then values from XQuery queries are assigned to specific attribute, for example to attribute with `Title` RDF name the mechanism assigns values from query `for $x in fn:doc({document})//msHeading/title return $x`. If there is a need for an attribute to have more that one query then line with additional query should be added (for example `Language` has two queries).

Wvery XQuery query should use `{document}` string to specify document on which the query is performed. Extension automatically replaces this string with appropriate path to XML file.

Appendix F. BibTeX metadata import

Editor/administrator application functionality may be extended with so called extensions. It is possible to add extension which imports metadata from external sources. BibTeX import extension is an extension of this type. It allows importing bibliographic description from BibTeX files (details about BibTeX can be found here).

In order to import bibliographic description BibTeX extension has to be properly configured.

BibTeX extension is configured using one property file (property files contain key=value pairs):

- `bibtexImport.properties` - file contains rules of conversion from BibTex format to *dLibra* metadata format.

Let us assume that we have the following `bibtexImport.properties` file:

```
Creator=book.author
Title=*.title;*.booktitle
Description=*.note
Publisher=*.publisher
Date=*.month;*.year
Type=*.type
Subject=*.keywords
Source=*.url
Contributor=
Identifier=
Language=
Relation=
Coverage=
Rights=
```

In the file key is attribute's RDF name and value is a list of fields (tags) of entries from BibTeX format. Concrete fields in the list of fields should be separated by the semilocon (`;`). For example (line 1) to `Creator` attribute values from `author` field, `book` entry will be assigned (notation `book.author`). If there is a need to import field values from any entry asterix character (`*`) should be used instead of entry name, for example (line 2) to `Title` attribute value from filed `title` and `booktitle`, any entry will be assigned (notation `*.title;*.booktitle`).

The following configuration is used as default:

```
Creator=*.author
Title=*.title;*.booktitle
Description=*.note
Publisher=*.publisher
Date=*.month;*.year
Type=*.type
Subject=*.keywords
Source=*.url
Contributor=
Identifier=
Language=
Relation=
Coverage=
Rights=
```

Appendix G. Attributes' values' Synonyms Dictionary

G.1. The idea of Synonyms Dictionary

The Synonyms Dictionary has been introduced in *dLibra* library in order to improve mechanism of searching library resources. User entering a phrase which has to be a key for searching may decide to extend the query by information included in dictionary. It gives them a better chance to find publications they looked for and on the other hand doesn't require for them to know the specific format of data used in attributes' values.

In example, let user input in a filed "author" value "Shakespeare" to search for library resources. If this query will be processed without synonymous extension only publications containing values with a character subset "Shakespeare" for an attribute "author" in bibliographic description will be found. However, if there is a group containing both values "Shakespeare" and "Szekspir" (polish translation of the name) for attribute "author", then user deciding to extend the query with a dictionary information will get as a result list of all publications that have "Shakespeare" or "Szekspir" set as an "author" value. This means that they don't have to know which format of this writer's name was used by librarians to describe his publications. The other examples of synonymous connection between attribute values may be: putting "picture" and "photo" in the same group for "format" attribute, or putting "polish", "polski" (polish translation of "polish" word), "pol" and "pl" in the same group for "language" attribute.

G.2. Construction of Synonyms Dictionary

dLibra system stores Synonyms Dictionary for attributes' values. This dictionary contains at least all those values used in bibliographic descriptions of library resources, but is not limited to those values and may contain phrases not used in a system. Separate dictionary is kept for each attribute and each language. dictionary contains at least all those values used in bibliographic descriptions of library resources, but is not limited to those values and may contain phrases not used in a system. Separate dictionary is kept for each attribute and each language.

For a given attribute and language the dictionary is divided into the groups of synonyms. Each group can contain one or more values. It is supposed that values belonging to the same group should either be synonyms, or words having the same meaning, or words in some other way connected with one another. In every group there is one marked value named base value. The group always takes it's name from it's base value. For a single attribute there may not exist more than one group having the same name. For a single group there may not exist more than one value having the same name, however values with the same name may exist in the separate groups for a single attribute.

Describing library element with a set of attributes means assigning these attributes concrete values taken from dictionary. One attribute may be assigned many different values. Such an approach causes that every modification of attribute value (for instance name change, deleting or moving value to a different group) influences bibliographic description of all resources being connected with this value. For example, if there are many Mickiewicz's publications in a system and all of them are connected with the same attribute's value, then fixing a mistake in this value will cause fixing description of all those publications.

Managing of Synonyms Dictionary is available in Library Manager application.

Appendix H. Editor's application default configuration

H.1. Default editor's application settings

Editor's application by default is set to simple mode (Section 3.4.1, “Editor's application modes (views)”) and adds automatically maila file format when new metadata are imported.

Appendix I. Objects' properties in *dLibra* system

Objects' properties avaliable in *dLibra* system has been described in the table below. One row contains the following pieces of information: *Element* column contains the name of an object, *Property* column contains the name of a property of an object, *Multilingual* column information whether the property is multilingual (*yes*) or not (*no*), *Required* column contains information which indicates wheter the property is required (*yes*) or not (*no*), *Description* column contains short description of the property.

Table I.1. Objects' properties

Element	Property	Multilingual	Required	Description
Directory	Name	No	Yes	Directory's name. Directories are only seen by editors in editor/administrator application.
Directory	Notes	No	No	Administrative notes, ie. information about elements in the directory. This information can only be viewed by editors.
Directory	Attributes values	Yes	No	Directory's bibliographic description is the default description for elements (such as publication) which will be created in this directory. When creating a new element in a directory user may provide bibliographic description of this new element. By default this description is filled with values from the directory. User may also modify this default description.
Directory	Identifier	-	-	Identifier cannot be modified as it is automatically assigned by the system. This is unique identifier of a directory.
Group publication	Name	No	Yes	Group publication's name in *dLibra* system.
Group publication	Notes	No	No	Administrative notes. This information can only be viewed by editors.
Group publication	Identifier	-	-	Identifier cannot be modified as it is automatically assigned by the system. This is unique identifier of a publication.
Group publication	Description	Yes	No	Short description of a group publication. This description is presented on web pages. It should contain short description of publications which are in the group publication.
Group publication	Comment	Yes	No	Comment to group publication. It may be expanded. It should contain additional information about the group publication.
Group publication	Miniature	No	No	Piktogram przedstawiający publikację grupową.

Element	Property	Multilingual	Required	Description
Group publication	Attributes values	Yes	No	Bibliographic description of a group publication is inherited by elements which are in this group publication. If bibliographic description of an element does not have values for specific attribute and group publication has such values then bibliographic description of the element will be filled up with the value from group publication. For every element such bibliographic description is automatically created when it is presented on WWW pages. Internally, in editor/administrator application bibliographic description of an element is separated from group publication's description.
Planned publication	Name	No	Yes	Planned publication's name in *dLibra* system.
Planned publication	Notes	No	No	Administrative notes. This information can only be viewed by editors.
Planned publication	Link	-	-	This value is automatically generated by the system. It is a permanent link to a publication in a digital library. It is composed of a “/publication/” part and publication's identifier, ie. for publication which have identifier “10” the link would be: “/publication/10”. In order to view information about the publication on WWW page the user have to run a browser (Internet Explorer, Firefox, etc.) and specify address composed of digital library address appended by the “Link”, ie. for a digital library which address is www.wbc.poznan.pl we would have: www.wbc.poznan.pl/publication/10.
Planned publication	Identifier	-	-	Identifier cannot be modified as it is automatically assigned by the system. This is unique identifier of a publication.
Planned publication	Attributes values	Yes	No	Bibliographic description of a planned publication is automatically assigned to first edition which is created while adding content to this planned publication.

Element	**Property**	**Multilingual**	**Required**	**Description**
Planned publication	Secured	No	Yes	This value is represented as check box. Editor may initially secure planned publication (see opis publikacji for more details) - mark, that the content of this publication should be secured.
Publication	Name	No	Yes	Publication's name in *dLibra* system.
Publication	Notes	No	No	Administrative notes. This information can only be viewed by editors.
Publication	Link	-	-	This value is automatically generated by the system. It is a permanent link to a publication in a digital library. It is composed of a “/publication/” part and publication's identifier, ie. for publication which have identifier “10” the link would be: “/publication/10”. In order to view information about the publication on WWW page the user have to run a browser (Internet Explorer, Firefox, etc.) and specify address composed of digital library address appended by the “Link”, ie. for a digital library which address is www.wbc.poznan.pl we would have: www.wbc.poznan.pl/publication/10.
Publication	Secured	No	Yes	This value is represented as a check box. If this check box is selected then the content of editions will be secured from copying and printing on WWW pages. This property may be checked only for HTML publications.
Publication without content	-	-	-	Publication without content has the same properties as normal publication. One difference is that the user cannot modify these properties (can only view them).
Edition	Name	No	Yes	Edition's name in *dLibra* system.
Edition	Notes	No	No	Administrative notes. This information can only be viewed by editors.
Edition	Published	No	Yes	This value is represented as a check box. If this check box is selected then users which have “view” right to the publication may view this edition. Edition may also be published until some date in the future - after this date the edition is automatically set to be not published.

Element	Property	Multilingual	Required	Description
Edition	Modification	-	-	Editor cannot modify this value - it is automatically updated by the system. The value is an identifier (login) of a user which has lately modified this edition.
Edition	Files' size	-	-	Editor cannot modify this value - it is automatically updated by the system. The value is total size of files which belong to this edition.
Edition	Attributes values	Yes	No	Bibliographic description of an edition. This description should correspond to the content of edition. Generally, every ediiton should have bibliographic description.
Edition	Description	Yes	No	Short description of an edtion. This description is presented on WWW pages. It should contain description of edition's content (short characteristic of the resource).
Edition	Comment	Yes	No	Comment to edition. It may be expanded. It should contain additional information about the resource, ie. information about damage on pages that were scanned.
Edition	Miniature	No	No	Icon which represents edition. It may be for example scan of first page of a book.
File	File name	-	-	File name cannot be modified. It is the name of a file which is a part of publication.
File	File type	-	-	File type cannot be modified. It is file type of a file which is a part of publication.
File version	File name	-	-	File name cannot be modified. It is the name of a file that this version refers to.
File version	File size	-	-	File size cannot be modified. It is the size of file version.
File version	Modified	-	-	This value cannot be modifies. It is last modification date of this file version.
File version	Type	-	-	Type cannot be modified. It is the type of file version.
File version	Description	No	No	Description of file version. It is an internal information for editors.

Element	Property	Multilingual	Required	Description
Collection	Name	Yes	Yes	Collection name in *dLibra* system. This name is presented on WWW pages.
Collection	Description	Yes	No	Collection's description. Description is presented on WWW pages.
Collection	OAI-PMH identifier	No	Yes	Identifier used in distributed resources search mechanizm. This identifier should be unique on one level of collections.
Collection	Notes	No	No	Administrative notes. This information can only be viewed by editors.
Collection	Identifier	-	-	Identifier cannot be modified as it is automatically assigned by the system. This is unique identifier of a collection.
User	Name	No	No	Data about the user (first name and surname).
User	e-mail	No	Yes	User's e-mail address. It must me unique (every user have to have different address).
User	Institution	No	No	Data about user (ie. institution where the user is employed).
User	Identifier	No	Yes	This value is provided once when creating user. When user is created it is not modifiable. This is unique identifier (login) of a user. User uses this identifier to log in to the system.
User	Password, confirm password	No	No	User's password. These two properties allow to change user's password.
User	User type	No	Yes	Value represented as a list. Types of user are described in ???.
User	Expiration date	No	No	After this date user account expires - user cannot log in to the system.
User	Blocked	No	Yes	Value represented as a check box. Account can be blocked by an administrator (when check box is selected). Blocked user cannot log in to the system.
User	Access domains	No	No	Access domains foe the user. It is possible to specify computer addresses that have access to the system without password or addresses that are blocked.
User	Groups	-	-	List of gruops that a user belongs to.

Element	**Property**	**Multilingual**	**Required**	**Description**
User	Administrative rights	-	-	Rights which may be assigned to a user. See ??? for more details.
Group	Name	No	Yes	Name of a group in *dLibra* system.
Group	Description	No	No	Description of a group.
Group	Members	-	-	List of users which are members of this group.
Group	Administrative rights	-	-	Rights which may be assigned to group. See ??? for more details.
Attribute	Name	Yes	Yes	Name of an attribute in *dLibra* system. This name is presented on WWW pages.
Attribute	Description	Yes	Yes	Description of an attribute in *dLibra* system. Description is presented on WWW pages.
Attribute	RDF name	No	Yes	Unique identifier of an attribute in *dLibra* system. It is used for example in specification of conversion rules for metadata import.
Attribute	Role	No	No	Role of an attribtue in Dublin Core schema.
Attribute	Identifier	No	Yes	Identifier cannot be modified as it is automatically assigned by the system. This is unique identifier of an attribute.

Appendix J. List of operations concerning objects of the *dLibra* system

Table below presents operations that may be performed on objects from editor/administrator application. Every row contains the following information: *Element* column contains the name of an object (element), *Operation* column contains the name of operation which may be performed on the element, *Access* column describes the place where the operation can be found and performed, *Description* column contains short description of the operation.

Table J.1. Operacje

Element	Operation	Access	Description
Directory	New directory	Management menu, toolbar, context menu	Operation allows editor to create a new directory in selected directory.
Directory	New publication	Management menu, toolbar, context menu	Operation allows editor to create a new publication in selected directory.
Directory	New group publication	Management menu, toolbar, context menu	Operation allows editor to create a new group publication in selected directory.
Directory	New planned publication	Management menu, toolbar, context menu	Operation allows editor to create a new planned publication in selected directory.
Directory	Delete	Management menu, toolbar, context menu	Operation allows user to remove selected directory including elements which are in it.
Group publication	New planned publication	Management menu, toolbar, context menu	Operation allows editor to create a new planned publication in selected group publication.
Group publication	New publication	Management menu, toolbar, context menu	Operation allows editor to create a new publication in selected group publication.
Group publication	New group publication	Management menu, toolbar, context menu	Operation allows editor to create a new group publication in selected group publication.
Group publication	Delete	Management menu, toolbar, context menu	Operation allows editor to remove selected group publication including publications which are in it.
Planned publication	Add content	Context menu	Operation allows editor to add content to planned publication. During this operation planned publication becomes normal.
Planned publication	Delete	Management menu, toolbar, context menu	Operation allows editor to remove selected planned publication.
Publication	Change files	Context menu	Operation allows editor to change files of publication. After this operation publication has completely new files (specified in wizard) - the old file are deleted. This operation is available when the publication has only one edition which is not published. Additionally, none of the publication files may have more than one version. These constraints are to make this operation available only in the erly stages of publication existence.

Element	Operation	Access	Description
Publication	New edition	Management menu, toolbar, context menu	Operation allows editor to create a new edition of the publication. By default, the newest files' version are selected to be the content of this new edition.
Publication	Download files	Avaliable only in advanced mode when publication has only one edition: Management menu, toolbar, context menu	Operation allows editor to download publication's edition files.
Publication	Delete content	Management menu, toolbar, context menu	Operation allows editor to delete content from all publication's editions. Before the operation is performed it is required to give a deletion reason. This reason is presented to WWW users which visit publication WWW page using permanent access point to publication.
Publication	Delete	Management menu, toolbar, context menu	Operation allows editor to delete publication. Before the operation is performed editor may give a deletion reason. This reason is presented to WWW users which visit publication WWW page using permanent access point to publication.
Publication without content	Delete	Management menu, toolbar, context menu	Operation allows editor to delete publication. Before the operation is performed editor may give a deletion reason. This reason is presented to WWW users which visit publication WWW page using permanent access point to publication.
Edition	New edition	Management menu, toolbar, context menu	Operation allows editor to create a new edition basis on selected edition. By default, in new edition wizard, files' versions which are grouped by selected edition are chosen.
Edition	Download files	Management menu, toolbar, context menu	Operation allows editor to download edition's files.
Edition	Delete	Management menu, toolbar, context menu	Operation allows editor to delete edition. Edition may be deleted when it is not the only edition of publication, it is not published and it is not edition of publication without content.

Element	Operation	Access	Description
Publication	Add new files' versions	Management menu, toolbar, context menu	Operation allows editor to add new files' versions to publication files. New files' versions are usually added in order to create a new edition composed of them.
Publication	Remove redundant files	context menu	Operation allows editor to organize publication files. More information concerning this operation can be found here.
File	Delete	Management menu, toolbar, context menu	Operation allows editor to delete file. File may be deleted when none of its versions belong to any edition.
File version	Download file	Management menu, toolbar, context menu	Operation allows editor to download file's version.
File version	Change file content	Context menu	Operation allows editor to change file's version content.
File version	Delete	Management menu, toolbar, context menu	Operation allows editor to delete file's version. File's version may be deleted when it does not belong to any edtion and it is not the only one version of a file.
Collection	New collection	Management menu, context menu	Operation allows editor to create a new collection in selected collection.
Collection	Delete	Management menu, context menu	Operation allows editor to delete selected collection.
User	New user	Management menu, context menu	Operation allows editor to create a new user.
User	Delete	Management menu, context menu	Operation allows editor to delete selected user.
Group	New group	Management menu, context menu	Operation allows editor to create a new group.
Group	Delete	Management menu, context menu	Operation allows editor to delete group.
Attribute	New attribute	Management menu, context menu	Operation allows editor to create a new attribute.
Attribute	Delete	Management menu, context menu	Operation allows editor to delete attribute.

Glossary

Publication	Digital object stored in a digital library.
LDAP	Lightweight Directory Access Protocol - protocol used for object localization (e.g. information about organization, user or file, etc.) which are grouped in a tree structure.
MARC 21 communitation format	File format which is used to store bibliographic description in MARC standard in order to communicate between systems.
RDF format	Resource Description Framework (RDF) - language used for representing information on the web