

# **User Guide for version 2.5 of the dLibra system**



**Poznań Supercomputing and Networking Center  
Edited by Tomasz Parkoła**

---

# **User Guide for version 2.5 of the *dLibra* system**

by Tomasz Parkoła

Copyright © 2002 - 2004 Poznań Supercomputing and Networking Center

---

# Table of Contents

|  |    |
|--|----|
| 1. Basic concepts .....  | 1  |
| 1.1. dLibra - general information .....  | 1  |
| 1.2. Three the most important dLibra users' types .....  | 1  |
| 1.3. Library structure - directories .....   | 2  |
| 1.4. Library collections - reader's view .....   | 2  |
| 1.5. Publication .....   | 2  |
| 1.5.1. File versioning .....   | 2  |
| 1.5.2. Publication editions .....  | 2  |
| 1.6. Object attributes .....   | 3  |
| 1.7. Other library resources .....   | 3  |
| 2. Program installation and startup .....  | 5  |
| 2.1. Installing and running dLibra GUI application .....   | 5  |
| 2.2. Logging into the dLibra GUI application and running editor or administrator application ..... | 6  |
| 3. Editor's Application .....  | 8  |
| 3.1. Editor's Application main window .....  | 8  |
| 3.1.1. Directories Tree .....  | 9  |
| 3.1.2. Elements list .....   | 9  |
| 3.1.3. Properties Window .....   | 10 |
| 3.1.4. Property editor .....   | 11 |
| 3.1.5. Bibliographic description editor .....  | 12 |
| 3.1.6. Rights Editor .....   | 16 |
| 3.1.7. Publication position editor .....   | 17 |
| 3.1.8. Assigning publication to collections .....  | 18 |
| 3.2. Publication Creator .....   | 20 |
| 3.2.1. Publication Creator's functions .....   | 20 |
| 3.2.2. Using Publication Creator .....   | 20 |
| 3.3. Typical editor's tasks .....  | 24 |
| 3.3.1. Creating/removing directory .....   | 24 |
| 3.3.2. Creating a new group publication .....  | 25 |
| 3.3.3. Creating a new publication .....  | 26 |
| 3.3.4. Creating a new planned publication .....  | 32 |
| 3.3.5. Adding content to planned publication .....   | 33 |
| 3.3.6. Publishing publication's edition .....  | 34 |
| 3.3.7. Deleting publication's content .....  | 34 |
| 3.3.8. Moving an object .....  | 35 |
| 3.3.9. Changing password .....   | 35 |
| 3.3.10. Searching .....  | 35 |
| 3.4. Advanced editor's tasks .....   | 35 |
| 3.4.1. Editor's application modes (views) .....  | 35 |
| 3.4.2. Downloading an edition of a publication .....   | 36 |
| 3.4.3. Adding new files' versions .....  | 38 |
| 3.4.4. Creating/removing an edition of a publication .....   | 39 |
| 3.4.5. Managing publication files .....  | 42 |
| 3.4.6. Mass publication upload .....   | 42 |
| 3.4.7. Editor's application configuration .....  | 46 |
| 4. Library Manager .....   | 49 |
| 4.1. Administrator's Application main window .....   | 49 |
| 4.1.1. Library Tree .....  | 49 |
| 4.1.2. Managing publications in collection .....   | 50 |
| 4.2. Basic administrator's tasks .....   | 52 |

|   |    |
|---|----|
| 4.2.1. Creating/removing collection .....                       | 52 |
| 4.2.2. Library attributes management .....                      | 53 |
| 4.2.3. Attribute's values dictionary management .....           | 56 |
| 4.2.4. Users and groups management .....                        | 60 |
| 4.3. Advanced right management .....                            | 68 |
| 4.3.1. Introduction .....                                       | 68 |
| 4.3.2. Library-level permissions .....                          | 68 |
| 4.3.3. Directory-level permissions .....                        | 69 |
| 4.3.4. Collection-level permissions .....                       | 71 |
| 4.3.5. Publication-level permissions .....                      | 72 |
| 4.4. Access domains management .....                            | 73 |
| A. MARC Import .....  | 75 |
| A.1. marcImport.properties configuration file format .....      | 75 |
| A.2. marcImpRemChars.properties configuration file format ..... | 77 |
| B. Import from and export to RDF format .....                   | 78 |
| B.1. RDF configuration files .....                              | 78 |
| B.1.1. namespaces.properties file .....                         | 78 |
| B.1.2. rdfImport.properties file .....                          | 79 |
| C. Attributes' values' Synonyms Dictionary .....                | 80 |
| C.1. The idea of Synonyms Dictionary .....                      | 80 |
| C.2. Construction of Synonyms Dictionary .....                  | 80 |
| D. Editor's application default configuration .....             | 82 |
| D.1. Default configuration files .....                          | 82 |
| D.2. Default editor's application settings .....                | 83 |
| Glossary .....  | 84 |

---

# Chapter 1. Basic concepts

## 1.1. *dLibra* - general information

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

- Storing digital documents of various types (e.g. scientific articles, reports, manuals, HTML pages, PDF documents)
- Managing digital documents stored in a library (e.g. modifying documents, grouping documents, assigning document to thematic categories)
- Access to digital documents (e.g. viewing document on web pages, searching for documents, securing documents from copying, temporal access)
- Managing a digital library (e.g. creating groups of users, managing user rights)

Editor application enables digital documents managing (e.g. adding, grouping documents), administrator application enables digital library management, whereas web application (WWW) enables browsing, searching and reading stored documents.

A set of discussion forums are available on *dLibra* project web page. There forums focus on 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. Three the most important *dLibra* users' types

To understand how does *dLibra* system work three common types of users have to be introduced:

- 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 documents (publications) to library, creates or imports bibliographic description for documents, groups documents, 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 documents (publications) 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. Library structure - directories

All the documents stored in the *dLibra* library are organized in a hierarchical structure of directories. Every directory can contain any number of documents 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. Library collections - reader's view

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 for *Biology* or *Computer science*).

## 1.5. Publication

A 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 comprise 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. File versioning

While developing an electronic document it very often is necessary to update or modify its content. As the previous versions of the publication may have already been published (made visible to the readers and library users) it is not possible to withdraw them from the library. Instead, new versions of the changed publication files can be stored in the library so that both the previous versions and the new ones are accessible. Creating a new version of a file does not, however, mean creating a new publication. The relationship between versioned publication files and the publication itself is covered in the next section.

### 1.5.2. Publication editions

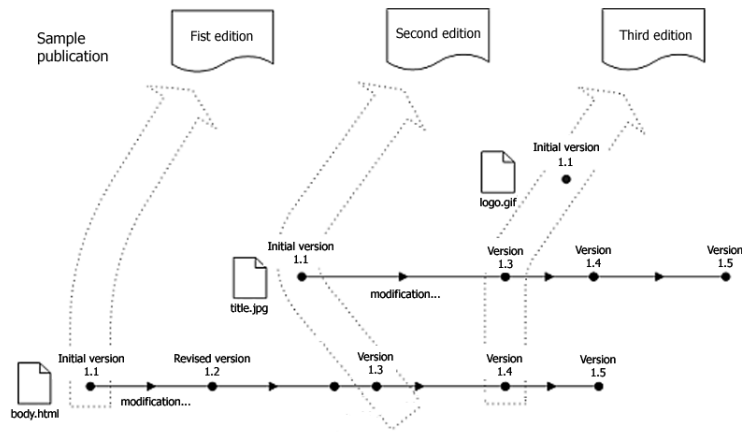
A publication can be made visible to the library users by creating an edition. An edition defines which versions of publication files will compose the document to be presented

to the user. Obviously, a single publication can have many editions including different versions of publication files or even different files.

To further investigate the idea of file versioning let us assume there is a publication consisting of only three files: `body.html`, `title.jpg` and `logo.gif`. The figure Figure 1.1, “File versioning and publication editions” shows how can subsequent versions of these files make publication editions.

In the figure, the sample publication starts from only one file: `body.html`. The first edition of the publication contains only this file, whereas the other files may or may not exist as well. Each publication object is versioned, which enables the editors to store subsequent revisions of the object and to place them in different publication editions. The second edition of the sample publication, apart from a revised `body.html` file, contains a new file: `title.jpg` in its initial version. The third edition comprises all the three files.

**Figure 1.1. File versioning and publication editions**



## 1.6. Object attributes

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 add or remove some of it's attributes. 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 bibliographic description. In case of default schema (Dublin Core 1.1) the description consist of information about author, title, publishing date, publisher, etc.

## 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 bibliographic description) is also a resource. Thus, attributes can be created and altered by the library administrators.



---

# Chapter 2. Program installation and startup

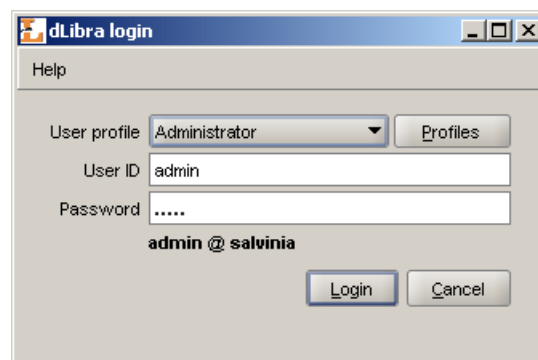
## 2.1. Installing and running *dLibra* GUI application

*dLibra* GUI application installation is maintained by *dLibra* library administrator. In order to run this application Java 2 Virtual Machine version 5.0 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 security question will appear --- press Always button.
6. On the screen login window should appear (Figure 2.1, "Login dialog").

**Figure 2.1. Login dialog**

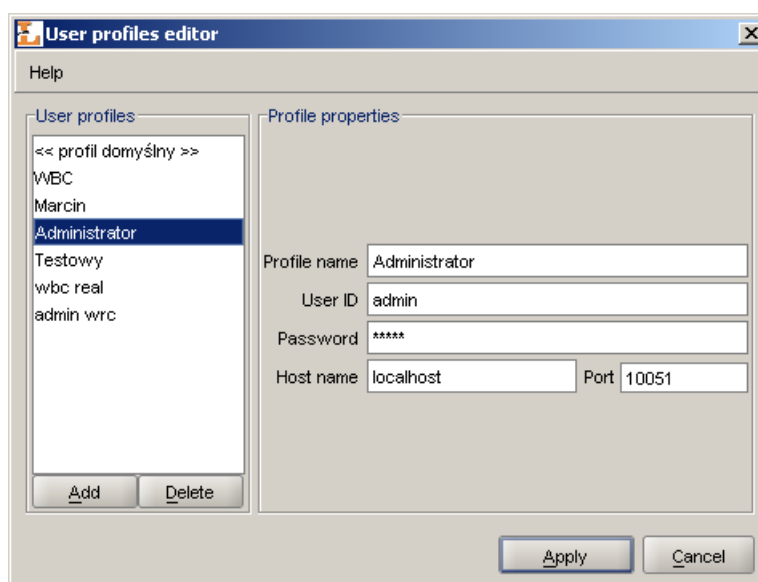


## 2.2. Logging into the *dLibra* GUI application and running editor or administrator application

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**



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.

After the successful 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**



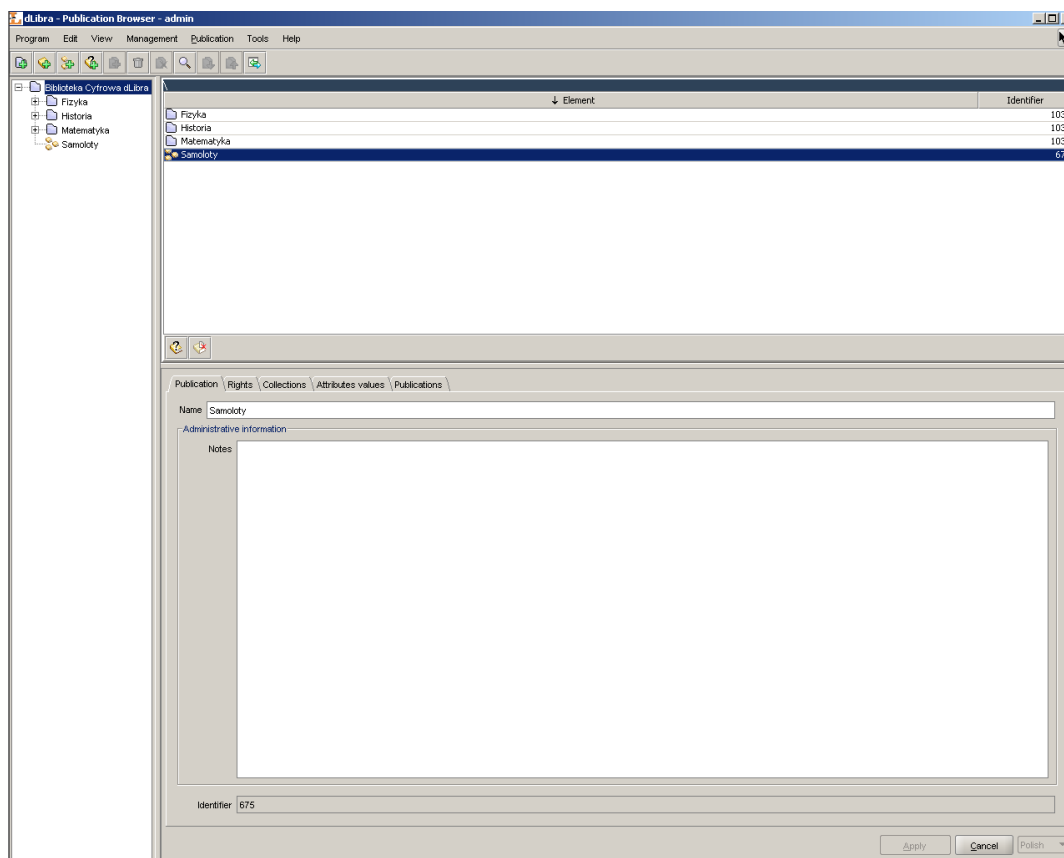
---

# 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



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 which selected object contains,
- Properties Window (on the right bottom part) describing element chosen on the list of elements. It contains tabs which present information about selected element.
- 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 (allows user to move elements).

In order to change component where by default elements list is placed 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, 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 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 is called by clicking a 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 main directory are listed on the elements list. You can open an elements on the list by doubleclicking on it or by pressing **ENTER** key (when element which we want to open is selected). The elements list is filled with elements which opened element contains. 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 an element is selected properties window (bottom left part of the window) is filled with information about selected element.

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.

Types of elements that the elements list may contain:

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

Two toggle buttons are placed on the bottom of the elements list. These buttons allow to hide specific elements. Currently, 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 in ????. In addition, publication in *dLibra* system may be secured. Only HTML (up to version 3.2 - higher versions may be incorrectly displayed) publications can be secured. Security mechanism disallows reader to copy, save or print publication content.

### 3.1.3. Properties Window

“Properties Window” enables modifying library element's properties. It consist of few (the number depends on the chosen element's type) properties editors placed on tabs. Pressing the Apply button will store the changed information (from every changed tab) on the server. Pressing the Cancel button or selecting another object on the elements list will discard all changes made since the last Apply operation. Values of some parameters (eg. edition name) can be defined in a number of languages - the current language can be changed with the drop down list placed on the bottom right corner of the “Properties Window”.

### 3.1.4. Property editor

Property editor enables viewing and modifying element's main properties, for example directory name for directory element. Property editor is the first tab in the properties window.

The list below shows properties for different 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 it, 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 light deleted):*
  - *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,
  - *Description* - description defines certain edition and contains basic information about it (eg. abstract) visible also in WWW application; edition description is language dependent and should be defined for each language separately,
  - *Comment* - comment contains additional information about the edition (eg. information about errors or damages in case of old books scans) visible also in WWW application; edition comment is language dependent and should be defined for each language separately,
  - *Notes* - administrative notes for an edition; notes content is language independent,

- *Miniature* - a picture representing a miniature of an edition; if miniature is chosen, it will be seen in web application.

Attention! Changing *Current Edition* properties has been disabled as it exists only for grouping the newest publication files versions of a given publication.

## **3.1.5. Bibliographic description editor**

### **3.1.5.1. Attributes values**

*dLibra* system provides an advanced attributes' values management mechanism to allow describing library elements with metadata. More information about the mechanism can be found in Appendix C, *Attributes' values' Synonyms Dictionary*. The idea of synonyms dictionary is described in MARC import appendix. In this section only basic information is presented about how the system works and how it influences the way user describes *directories*, *group publications* and *editions* with metadata.

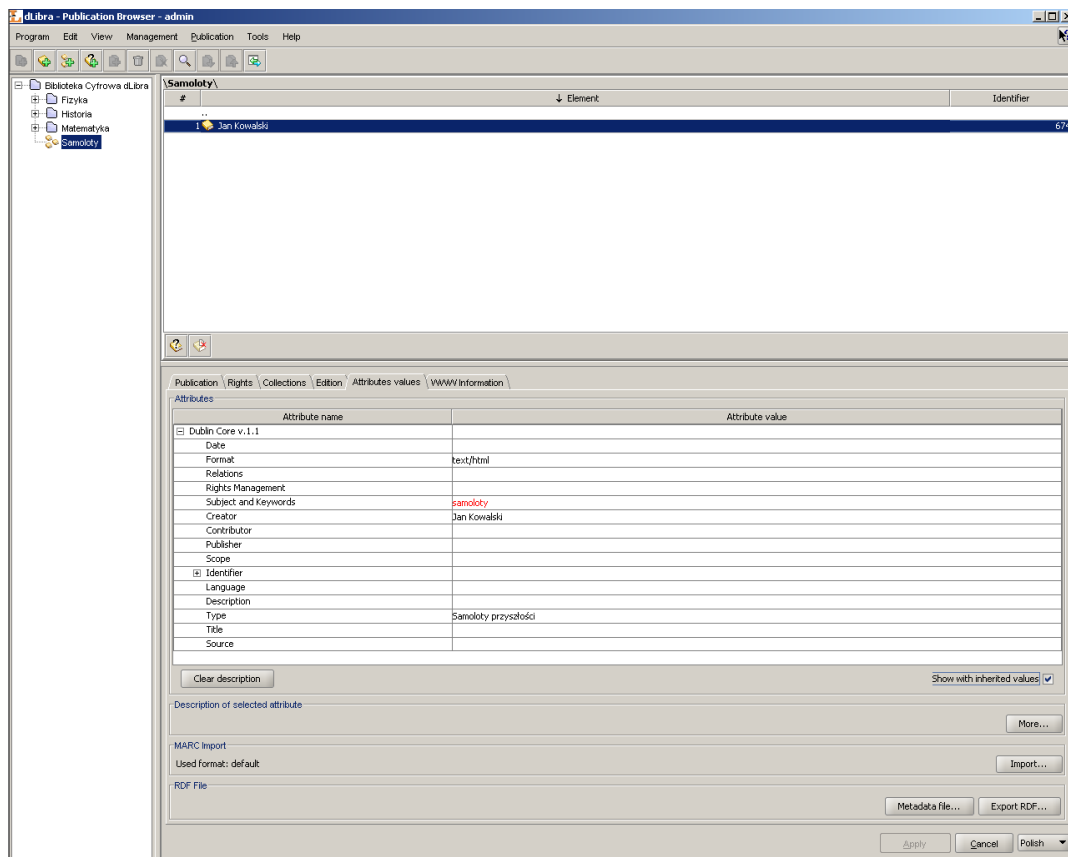
Attributes' values are kept in a system in a special dictionary. It is built separately for each language and each attribute. Therefore it is allowed for a value to exist more than once, singly for each attribute. Values of every attribute are divided into groups of synonyms. The same value may exist in many groups, for instance if it has many meanings. Describing library resources with metadata rest on choosing a set of values for some attributes. If a value doesn't exist in a dictionary, it is possible to add it using Publication Browser.

### **3.1.5.2. Attributes' Values Editor**

In the Attributes' Values Editor (???) object's attributes can be altered. In *dLibra* system set of attributes may describe *directory*, *edition* and *group publication*. Bibliographic description of an edition is essential for searching the resources based on their attribute values. 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 description for all newly created editions of any publication in this directory, as well as for group publication for which this directory is a direct parent.



**Figure 3.2. Attributes' values editor**



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

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).

Under the table with bibliographic description, on the right side, Show with inherited values check box is placed. When this check box is selected the table with bibliographic description is fulfilled with attributes' values inherited from overriding group publications. Also under the table with bibliographic description, but on the left side, Clear values button is placed which removes all attributes' value from the table.

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

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

2. Enter the value you want to add,
3. Approve the value by pressing the **ENTER** key.

In order to add a value to an attribute when there is at least one value assigned to the attribute:

1. Choose a field with value assigned to the attribute to which you want to add the next value,
2. Press the **TAB** key which will add a new row, below previously selected field,
3. Enter the value you want to add,
4. Approve the value by pressing the **ENTER** key.

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

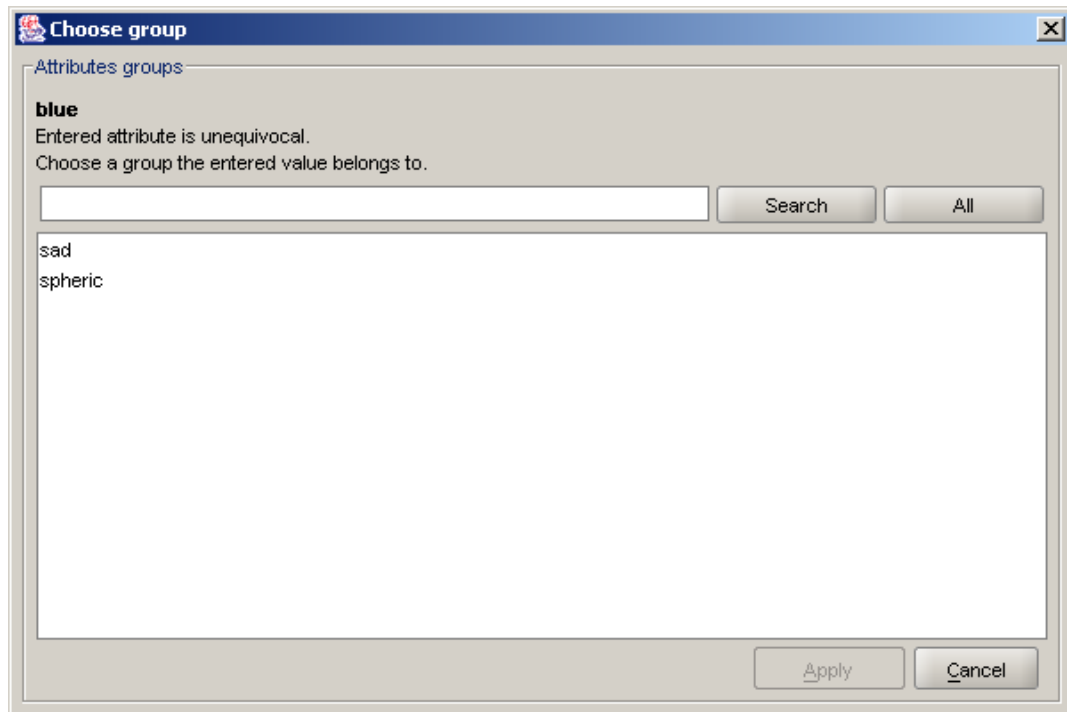
If the attribute to which the value which is being entered has already defined values and these values start with the same characters which are being entered then autocomplete mechanism will be started. This mechanism shows the list of attribute's values (see Figure 3.3, "Attribute's values list shown by the autocomplete mechanism") which start with already entered characters. If the value to enter is on the list it is possible to select it directly (e.g. by double clicinkg on its name).

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

| Attribute name                             | Attribute value             |
|--|-----------------------------|
| <input type="checkbox"/> Dublin Core v.1.1 |                             |
| Title                                      | Ne                          |
| Creator                                    | New theories in physics     |
| Subject and Keywords                       | Never again - be pefrect    |
| Publisher                                  | New adventures of blue bear |
| Contributor                                |                             |
| Resource Type                              |                             |
| Format                                     |                             |
| Resource Identifier                        |                             |
| Source                                     |                             |
| Language                                   |                             |
| Relation                                   |                             |
| Coverage                                   |                             |
| Rights Management                          |                             |
| Description                                |                             |
| Date                                       |                             |

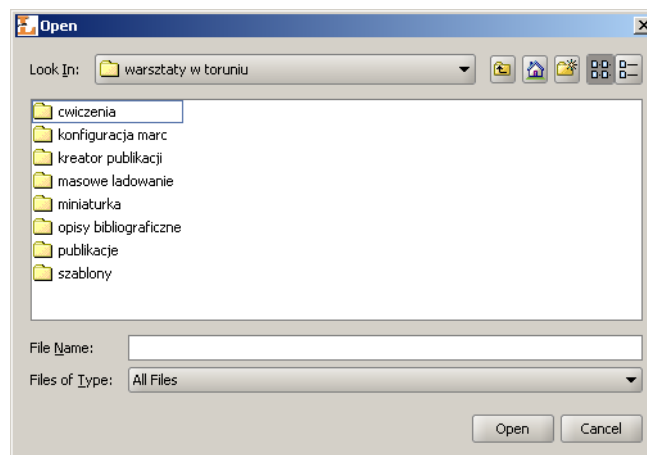
Attention! In order to save attributes' values (bibliographic description) press Apply button in the bottom right corner of the "Properties Window".

During the apply operation, for every attribute's value it is checked if there already exists such a group on the server. If it does not exist --- new group is created. If a group exists and it is only one such a group --- this group is used. If there are more than one group then it is needed to choose the group by the user. The choose group window (see Figure 3.4, "Group selection for a value") is presented in such equivocal situation and enables user to choose the group to which entered value should be assigned.

**Figure 3.4. Group selection for a value**

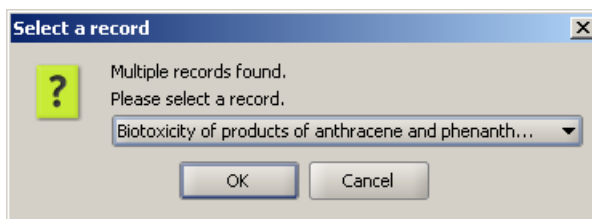
### 3.1.5.3. Import from MARC format

*dLibra* allows user to import attributes' values directly from a document in *MARC 21 communication format*. An interface grouped in MARC Import panel (placed in “Bibliographic description editor” panel) was prepared for this feature. Import format is fully configurable for the user. A text information on this panel shows a path to a directory (or link) containing import format used in application. A word *default* stays for default built-in application format. MARC import settings can be changes in application configuration. Import... button enables choosing file (Figure 3.5, “Choose file window”) which is to be imported. You have to remember that a file which is to be imported has to saved in *MARC 21 communication format*.

**Figure 3.5. Choose file window**

If the file which is to be imported encloses more than one MARC record, the application will ask the user to choose one (Figure 3.6, “Choose MARC record window”).

**Figure 3.6. Choose MARC record window**



#### **3.1.5.4. RDF format - import and export**

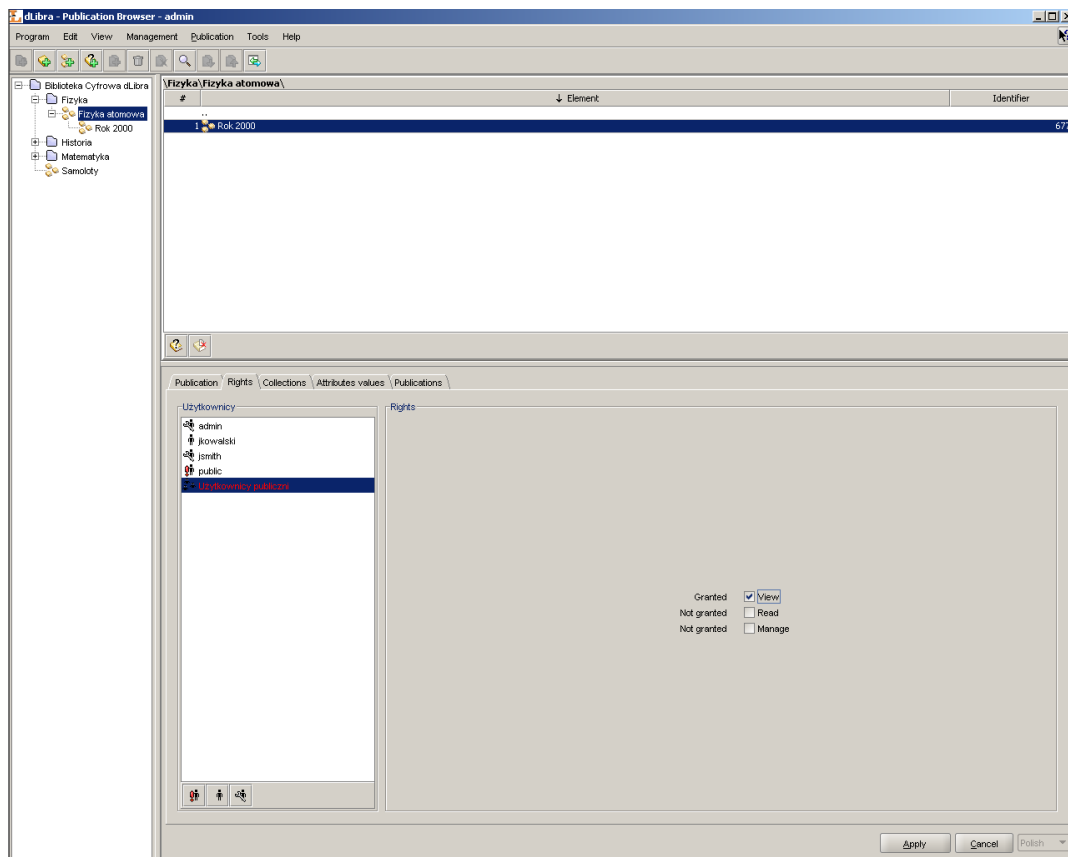
*dLibra* enables importing and exporting attribute's values directly from or to *RDF format* file. Import RDF panel (placed in “Bibliographic description editor” panel) is dedicated to import *RDF format* files. Import format is fully configurable for the user. RDF import and export settings can be changed in application configuration. Metadata file... button enables choosing file (Figure 3.5, “Choose file window”), which is to be imported. You have to remember that a file which is to be imported has to be saved in *RDF format*. Export RDF... button allows saving RDF file which will contain bibliographic description from bibliographic description editor.

#### **3.1.6. Rights Editor**

The Right Editor (Figure 3.7, “Right Editor - publication rights”) enables to alter access rights to library objects. In the Editor's Application access permissions to a publication can be changed.

For a publication as a whole three kinds of rights can be granted:

- *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 3.7. Right Editor - publication rights**

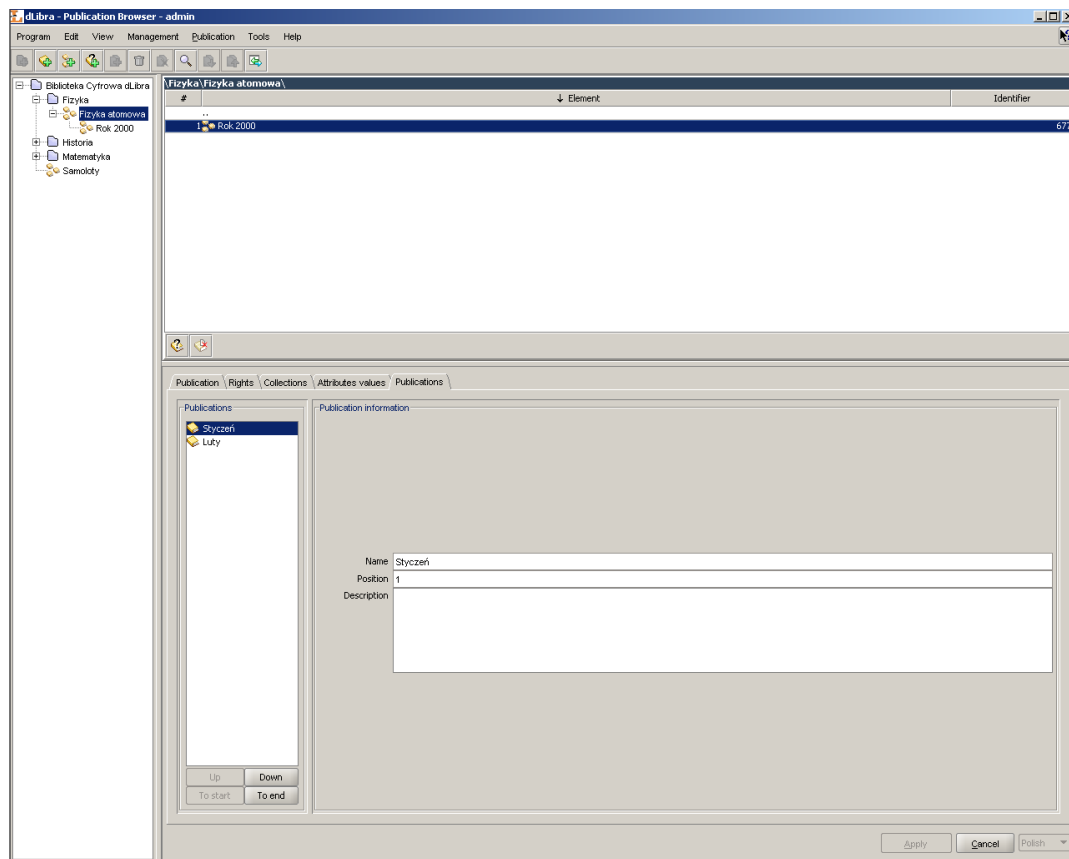
Every right, regardless of the object in the library it refers to, can have one of five states:

- *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 (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”).

### 3.1.7. Publication position editor

In a group publication its every component has assigned a position on which it appears. By default, newly added publication is said to be on the last position. It is possible to manipulate publication position using publication position editor (Figure 3.8, “Publication position editor - publication positioning”).

**Figure 3.8. Publication position editor - publication positioning**

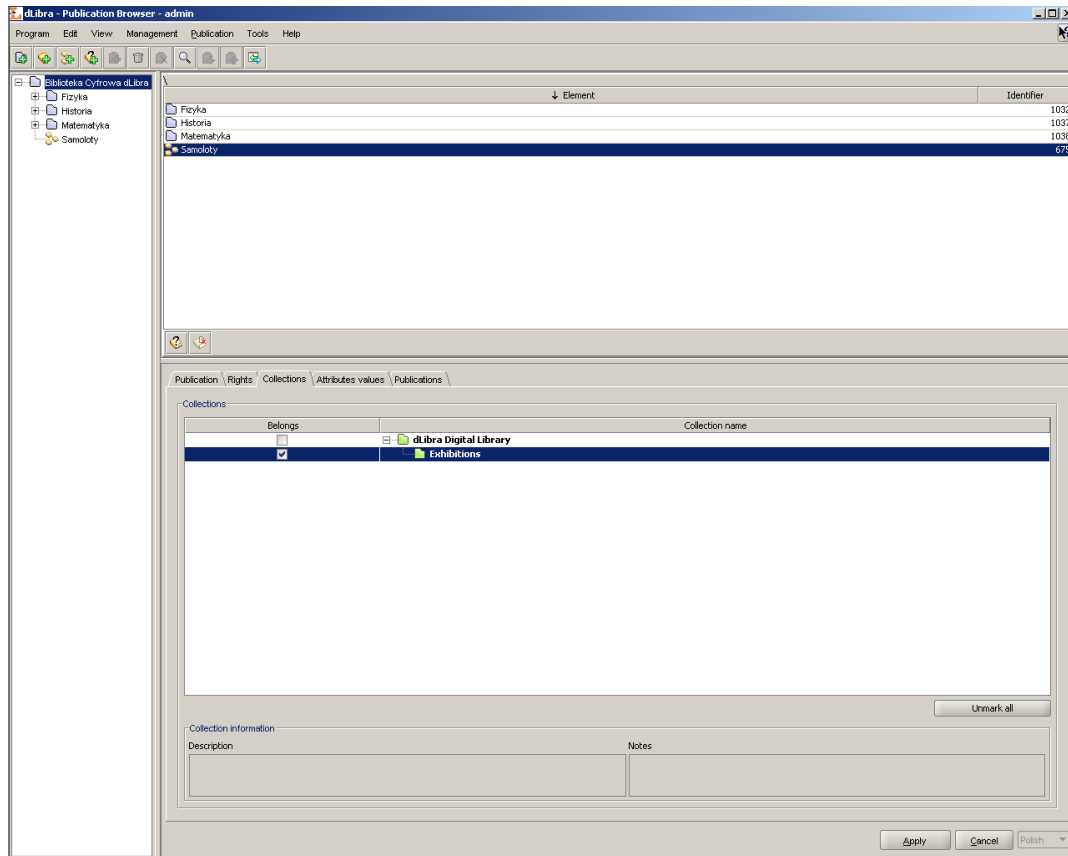
In the editor's left part, on the Publications panel there is a list of publications which are element of currently checked (on the elements list) group publication. Under this list there is a set of buttons to manipulate publication position. In the editor's right part (Publication information panel) are shown basic information about selected publication (on Publications panel). To change selected publication position use Up, Down, To start or To end button.

### 3.1.8. Assigning publication to collections



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.

For each publication user may choose collections this publication should belong to. An interface enabling it is placed on a tab named Collections allowed after choosing a publication on elements list. The same interface is also prepared as one step of new publication wizard.

**Figure 3.9. Adding publication to collections**



At the top of the panel there is a table containing two columns. Right column contain collections tree, which shows the structure of all collections in the system. Left column contain checkbox showing if publication belongs to the collection or not. After choosing a single publication on elements list on the left side of Editor's Application, all collections that chosen publication belongs to are marked with checkbox in a collections table.

If collection icon is a blue folder  it means that currently logged on user has rights to manage this collection and may change the membership state of a publication (if publication belongs to the collection they may remove it, if not they may add it to collection). Red folder icon  means that user has not appropriate rights to change membership state of the publication in a chosen collection.

If a collection name in a collections tree is marked with bold font it means that chosen publication will belong to this collection (directly or indirectly). Checking each collection causes the marking of this collection name and all supercollections' names on the path to the main library collection. All changes made must be accepted by clicking Apply button on the bottom of Properties Window. Choosing different object or clicking Cancel causes canceling of all changes made since last apply action.

Collection description and notes, available on the bottom of panel, are shown only as an information and they cannot be edited in there.

## 3.2. Publication Creator

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

### 3.2.1. Publication Creator's functions

Publication Creator is designed to help preparing publications in HTML format, with files shown by user. It 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 into the library. If yes, new publication creator is started. The first step is selecting destination (either directory or group publication), the remaining steps are the same as for new publication creator. As a main file it is required to select generated by creator `index.html` file.

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.).

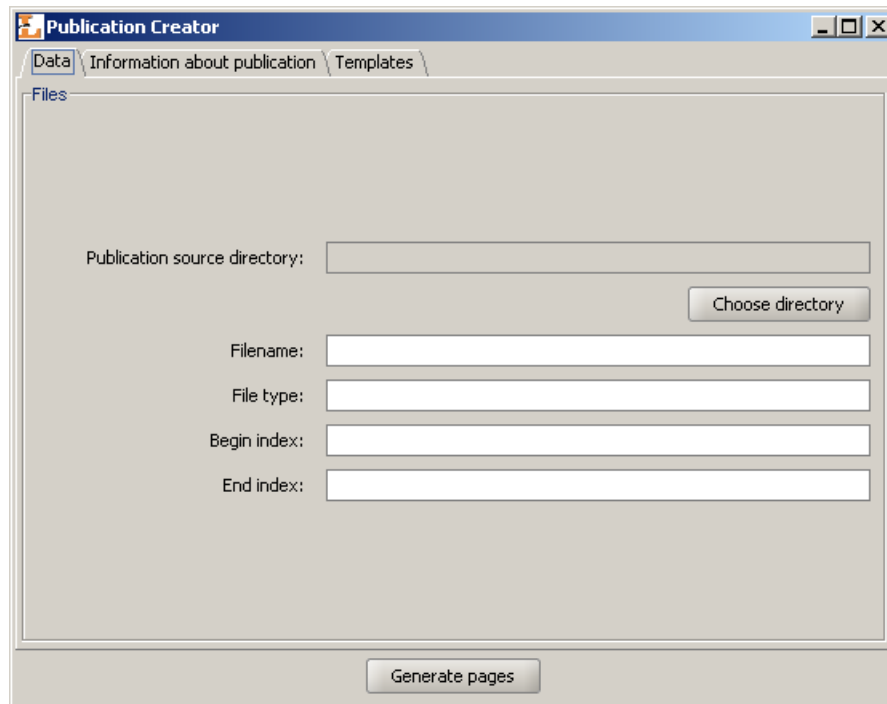
### 3.2.2. Using Publication Creator

Publication Creator interface is divided into three tabs: Data, Information about publication and Templates. Down the tab panel there is a Generate pages button. After pressing the button, publication will be created in a chosen directory.



### 3.2.2.1. "Data" tab

Figure 3.10. Publication Creator - Information about files tab



On a Data tab there is placed an interface allowing to choose the files that should be used to generate a publication, and to choose a directory in which the publication should be saved. In a Files panel user should choose a directory containing files. A Choose directory button is designed to allow this. After pressing it user must select one of the files that will be used to create a publication. After doing so, fields in a File panel will become filled with a default 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 into a directory where the selected files are (this directory is pointed by Publication source directory field).

### 3.2.2.2. "Information about publication" tab

Figure 3.11. Publication Creator - Information about publication tab

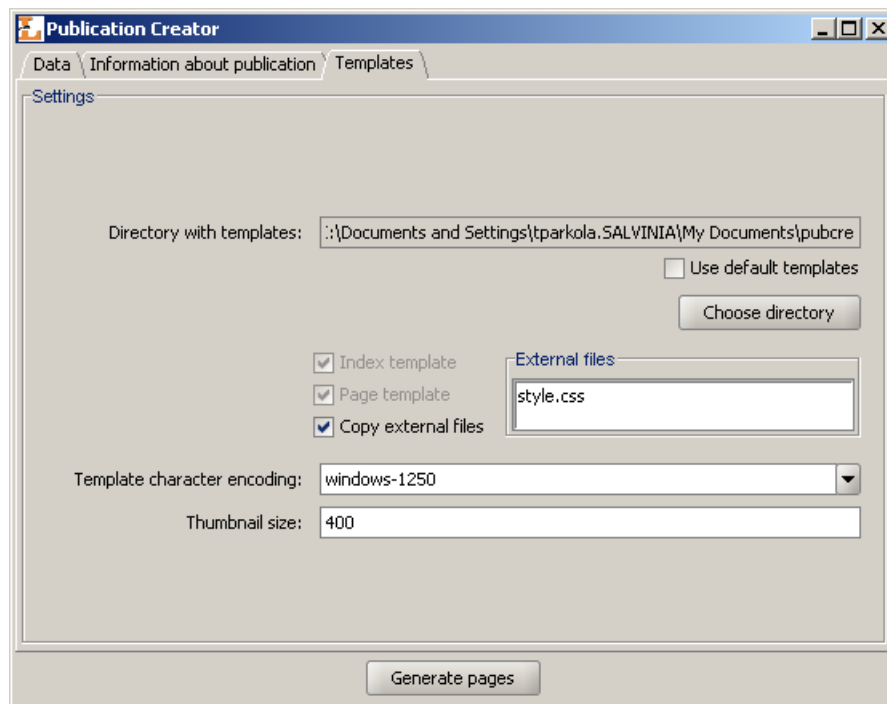
The screenshot shows a window titled "Publication Creator" with three tabs: "Data", "Information about publication", and "Templates". The "Information about publication" tab is selected. It contains two sections: "Basic information" and "Additional information (HTML)". Under "Basic information", there are two text input fields: "Author:" with the value "John Smith" and "Title:" with the value "Title". The "Additional information (HTML)" section is a large empty text area. At the bottom center of the window is a "Generate pages" button.

On a Information about publication tab there is placed an interface allowing to input some additional information about publication. This information will be placed in the HTML files and include:

- Author - Information about publication's author
- Title - A title of created publication
- Additional information - Additional information about publication, inputed in an HTML format.

### 3.2.2.3. "Templates" tab

Figure 3.12. Publication Creator - Information about templates tab



On a Templates tab there is placed an interface allowing to define what templates must be used to generate a publication. To choose templates user must press Choose directory button and choose a directory in which templates are stored. After choosing a directory, two disabled check boxes will change their state, depending on whether templates for page and index were found. Index template should have a `index.vm` name, and page template should have a `page.vm` name. On an External files list there will be shown all additional files that are used by the templates. Selecting Copy external files check box will cause that after generation process those files will be copied to target publication directory. The last things that should be defined are template character encoding and size of thumbnails that will be generated (useful only when generating publication composed of pictures). It is also possible to use default templates. To do it, simply check the Use default templates box.

Templates chosen on Templates tab must have a format compatible with Velocity standard. They may have the 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 go to the Velocity homepage: <http://jakarta.apache.org/velocity/user-guide.html>.


## 3.3. Typical editor's tasks

This section describes how to perform typical editor tasks using *dLibra* “Editor's application”.

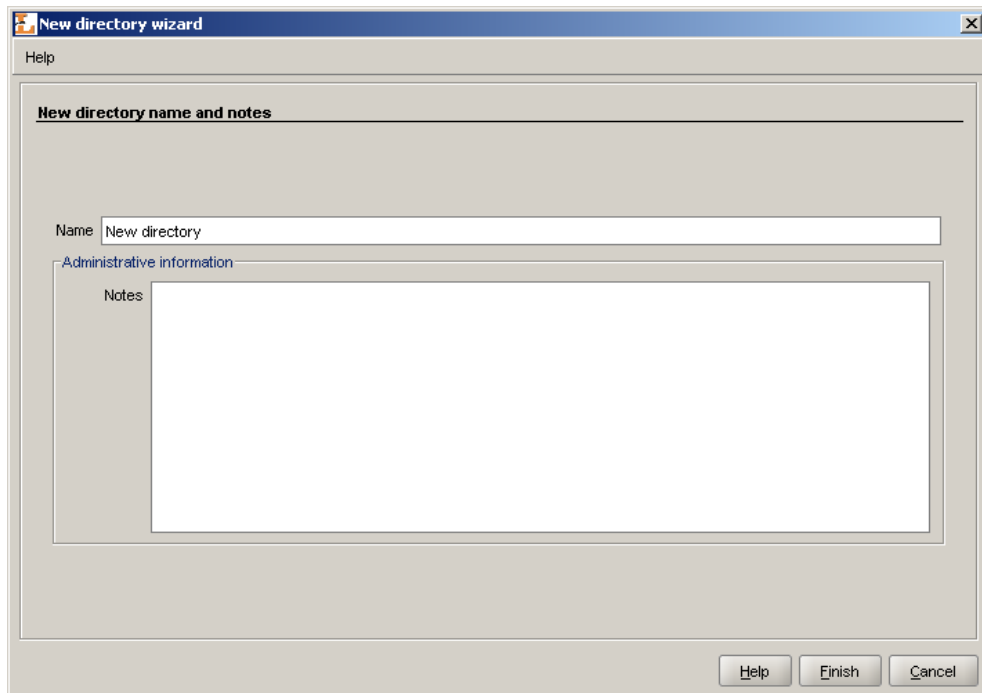
### 3.3.1. Creating/removing directory

#### 3.3.1.1. Creating, moving and removing library directories


To create a new directory:

1. On the elements list select the library directory to which the newly created directory should belong.
2. From the popup menu select the New directory function.... Alternatively, the toolbar button  or main menu options can be used.
3. In the New Directory Wizard enter name and description for the new directory. Press Finish button to create the directory.

**Figure 3.13. New Directory Wizard**



To remove a directory:


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

If the directory contains at least one publication it is possible to specify remove reason for all the publications contained in the directory.

To move a directory or a publication from one location to another - use drag-and-drop method or Cut/Paste options from the context menu.

### **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 main menu options can be used.
3. Enter the name and notes for the new group publication and click Finish button to finish.

### 3.3.3. Creating a new publication

**Note**

Before a new publication can be placed in the library all its files must be present in a directory on a local disk.

To create a new publication:

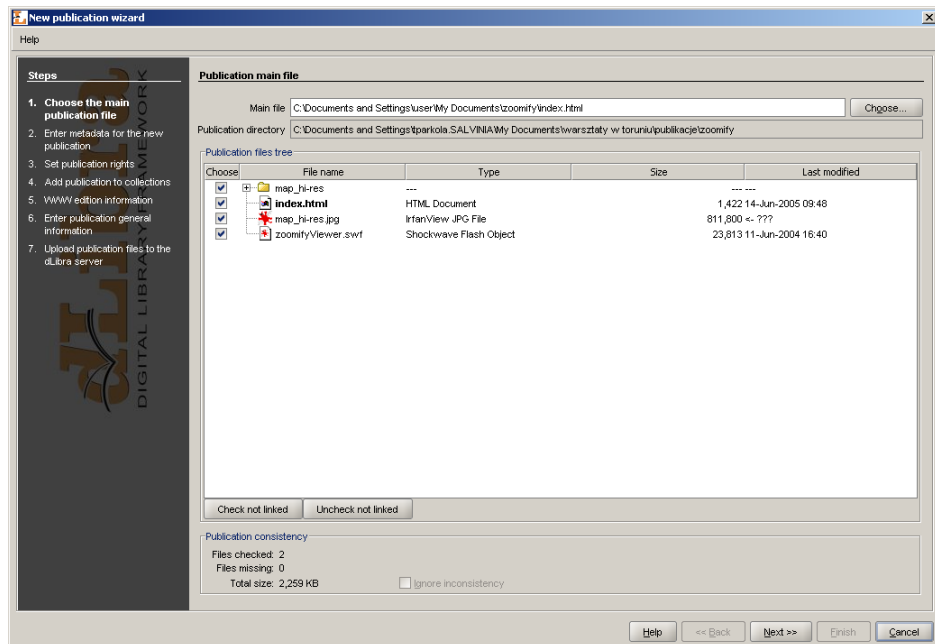
1. On the elements list select the directory or group publication the new publication should belong to.
2. From the popup menu select the New publication.. function. Alternatively, the toolbar  button or main menu options can be used.
3. In the first step of the New Publication Wizard (Figure 3.14, “New Publication Wizard - main publication file page” ) select the main publication file. If the publication consists of one file such as a PDF or a MS Word document the file should be selected. In case the publication is in the HTML or DJVU format the top level (e.g. the `index.html`) file should be selected.

If the main publication file is an HTML or DJVU file the wizard will check whether all the linked files are present. 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 publication. All files publication consists of will be added to publication files table. There is no way to uncheck them, they all will be sent to server.

Besides files found by the creator, in the files table all other files (which are in the directory tree of the directory which encloses the main file) will be placed. By default these files are not checked to send to the server. It is possible then to add files which were not found by the creator, but belong to the publication.

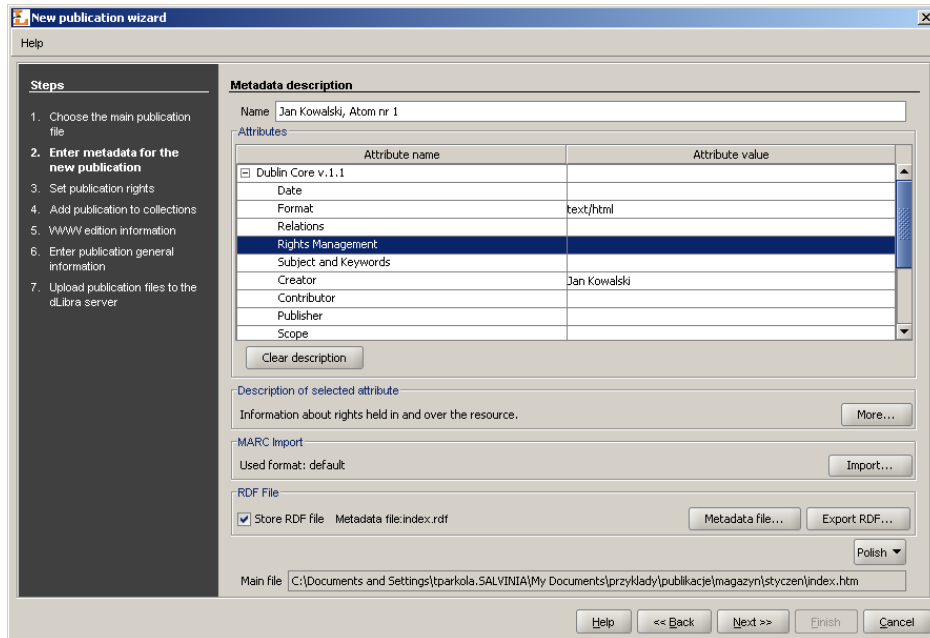
Notice that main publication file cannot be unchecked and it will always be sent to the server with application being uploaded.

Press the Next button to move to the the page.

**Figure 3.14. New Publication Wizard - main publication file page**

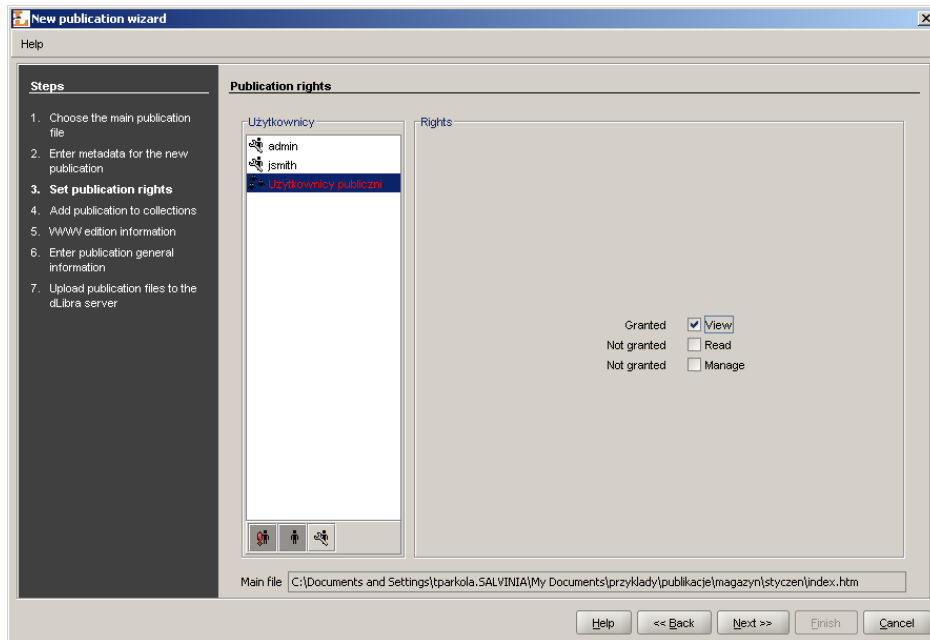
- In the second step ( Figure 3.15, “New Publication Wizard - attribute values page”) publication may be described with a set of attributes. Bibliographic description may be imported from a file saved in *MARC 21 communication format* or *RDF format*. The way for defining attributes' values and importing from both formats is described in the part concerning Bibliographic description editor. In order to proceed to next step a publication name has to be given (field with Name label). By default the name is created basis on bibliographic description through combining the title, author and publishing date. 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 a user changes the attributes' values the name is updated. Press the Next button to move to the next steps.

**Figure 3.15. New Publication Wizard - attribute values page**



5. In the third step ( Figure 3.16, “New Publication Wizard - publication rights page” ) user can add appropriate rights to created publication. For more details about rights management for publication see section Rights editor. Press the Next button to move to the next steps.

**Figure 3.16. New Publication Wizard - publication rights page**

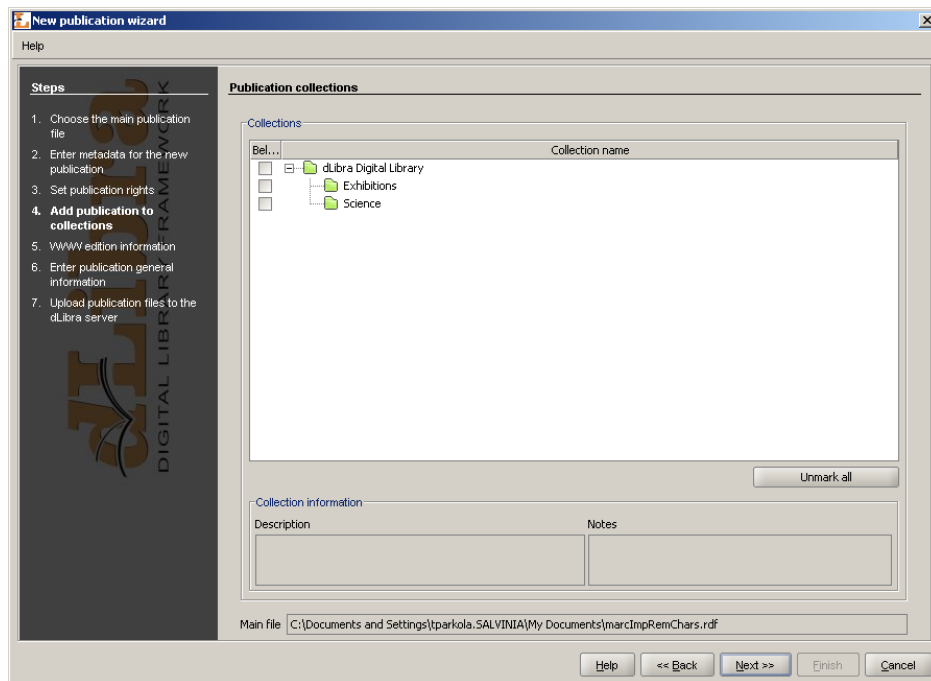


6. In the fourth step (Figure 3.17, “New Publication Wizard - publication's collections page”) it is possible to assign publication to collections. This mechanism is exactly



the same as described in Publication's collections editor. Press the Next button to move to the next step.

**Figure 3.17. New Publication Wizard - publication's collections page**

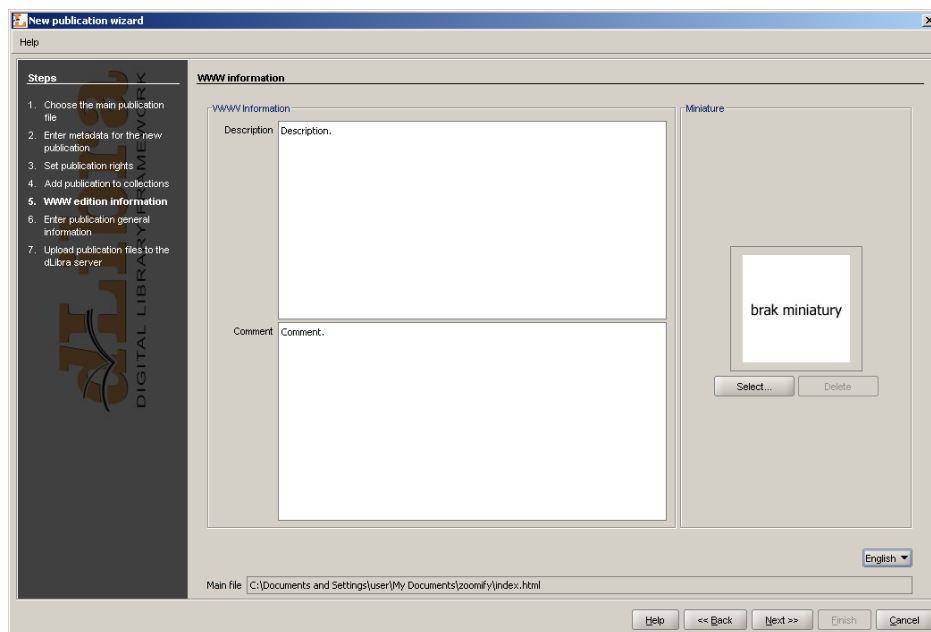


7. In the fifth step (Figure 3.18, “New publication wizard - WWW information”) it is possible to specify WWW information concerning the first edition of the new publication. WWW information consists of:

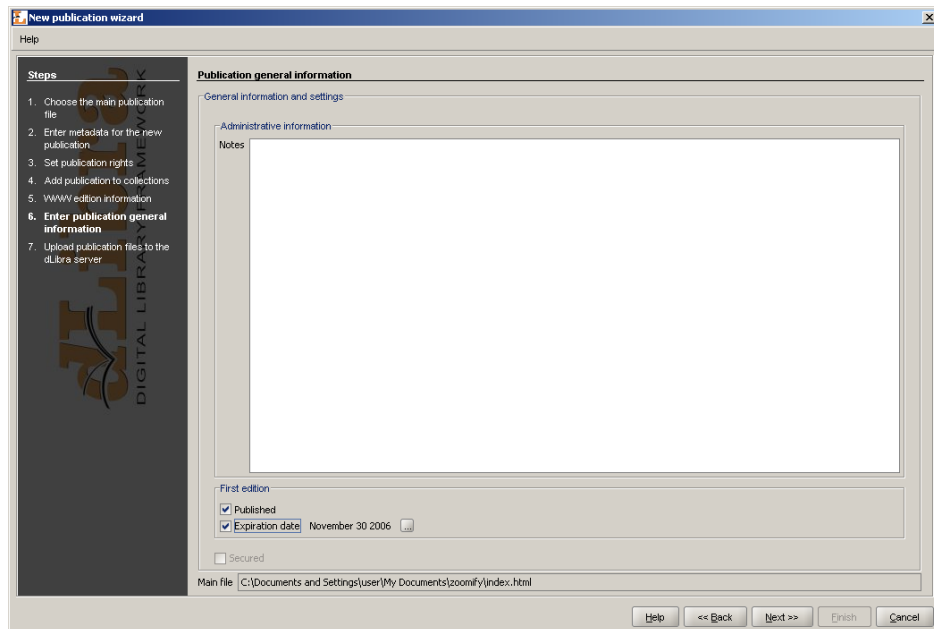
- Description - short edition description, e.g. what does it concern, visible near search results on the web application (in order to show user only important notes about the edition it is recommended to provide the description possibly short --- one or two sentences),
- Comment - characteristic information concerning the edition, e.g. specific marks (for example lack of one page),
- Miniature - graphic edition representation visible by the web application users near information about the edition.

Press the Next button to move to the next step.

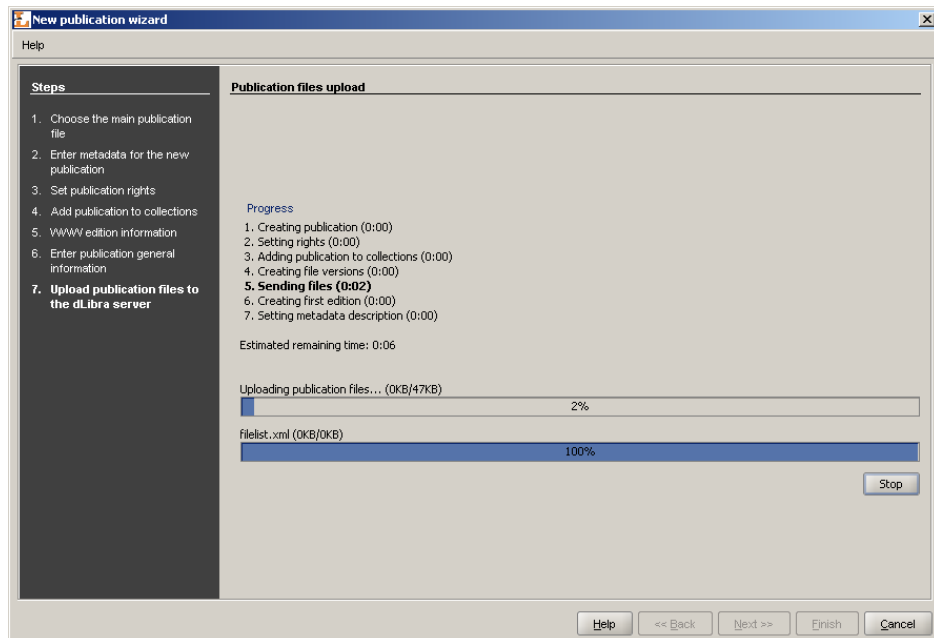
**Figure 3.18. New publication wizard - WWW information**



8. Step sixth ( Figure 3.19, “New publication wizard - general information”) enables providing three pieces of information:
  - Notes - administrative notes concerning publication
  - First edition - panel which allows for publishing the first edition (uncertain or time publishing)
  - Secured - check box which allows for securing publication from copying (applies only to HTML publications)

**Figure 3.19. New publication wizard - general information**

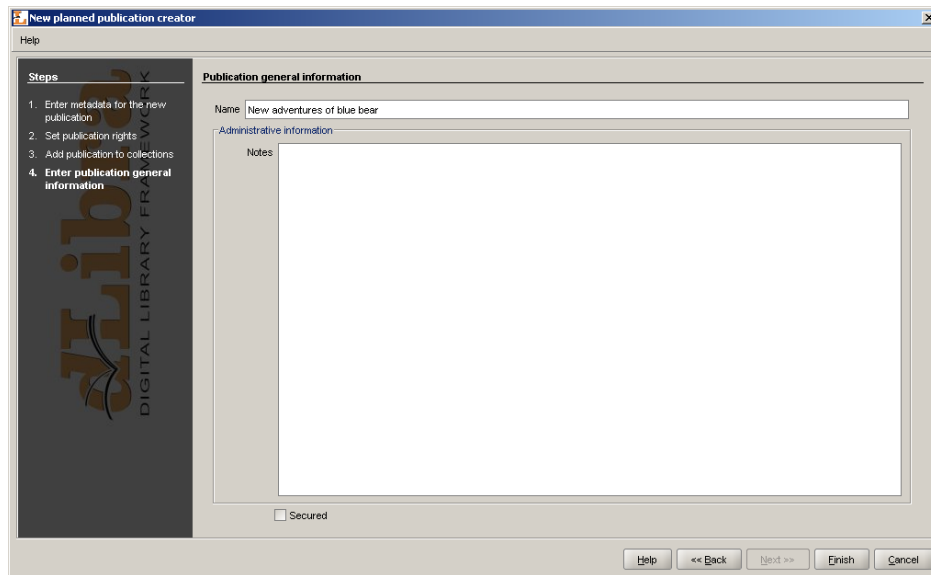
9. Uploading files on the server is the last step (Figure 3.20, “New publication wizard - creating publication on the server”). Press the Start button to create publication on the server. The process can be paused and resumed pressing the button again. During the creation process on the Progress panel all operation which have to be performed are presented. Operation which is being performed is bolded. For every operation elapse time is measured and presented (in seconds) in the brackets after the operation name. Progress bars inform about progress of currently performed operation. When all operations are finished short summary is presented. During the creation process it is possible to cancel it (e.g. press the Cancel button).

**Figure 3.20. New publication wizard - creating publication on the server**

### 3.3.4. Creating a new planned publication

To create a new planned publication

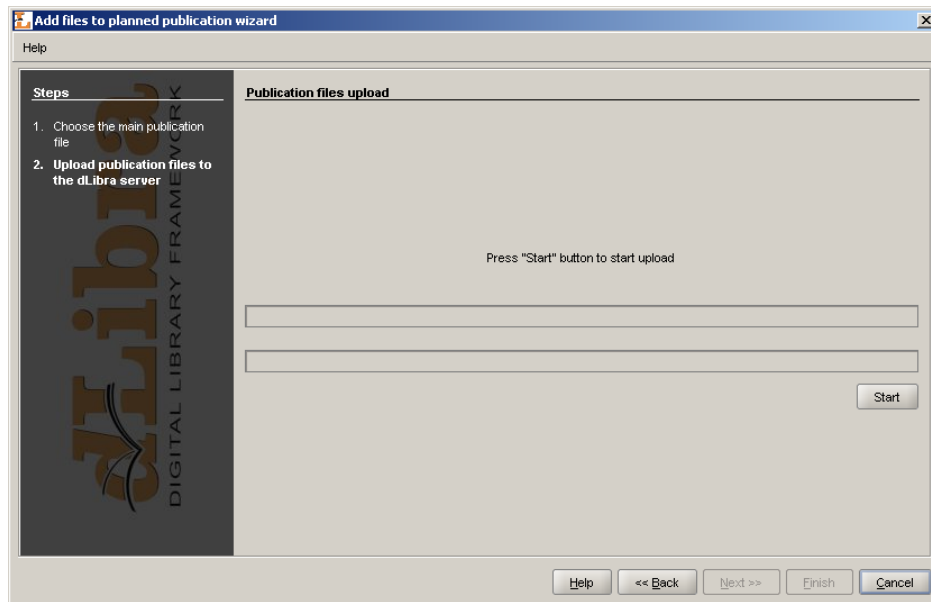
1. On the elements list select directory or group publication which new planned publication is supposed to be in
2. From the popup menu select New planned publication.... It is also possible to select this option from toolbar or main menu.
3. Planned publication wizard consist of four steps. The first 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 3.21, "New planned publication wizard - general information" ) specify the administrative notes and information whether the publication should be secured or not (Secured check box).

**Figure 3.21. New planned publication wizard - general information**

### 3.3.5. Adding content to planned publication

To add files to planned publication:

1. On the elements list select a planned publication which we want to add files to.
2. From the popup menu select Add files....
3. Add content wizard consists of two steps. The first step is the same as the first step of new publication wizard. The second step ( Figure 3.22, “Add content wizard - sending content (files) on the server”) allows for sending publication content on the server. In order to send selected files in the first step press Start button. Sending files on the server progress is presented on progress bars.

**Figure 3.22. Add content wizard - sending content (files) on the server**

### 3.3.6. 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.

### 3.3.7. Deleting publication's content

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

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 the reason for removal. Attention! The reason is required in order to delete publication's content because it is usefull when informing readers that the publication was removed.

### **3.3.8. Moving an object**

There are two ways for moving an object. First way is possible when editor application has two elements list visible (see???). In order to move selected element in 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. 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.

### **3.3.10. Searching**

User may find an element in two different ways.

- Searching for element basis on bibliographic description,
- searching for element basis on its identifier.

Searching basis on bibliographic description gives the same results as searching on WWW. In order to run this searching select Find... option from Edit menu.

To run searching basis on identifier select Go to... option from Edit menu. Then select type of element to search and type identifier to search. If such an element exists it will be displayed and selected on elements list.

## **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**

| <i>Feature \ Mode</i>   | <i>Simple mode</i>  | <i>Advanced mode</i>   |
|-------------------------|---|--|
| <i>Publication node</i> | <ul style="list-style-type: none"> <li>• If the publication has only one edition it is not visible --- after expanding the publication, edition's files versions are visible. All the properties for the edition and publication can be specified on the tabs in the properties window,</li> <li>• If the publication has more than one edition --- these editions are visible after expanding the publication (it is the same situation as in the advanced mode).</li> </ul> | After expanding publication all its editions are visible (including the <i>current</i> one). Files' version which compose the edition are visible after expanding edition. |

### 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 3.23, "Download Wizard - publication file selection page") select the local disk directory the publication files shall be stored in.

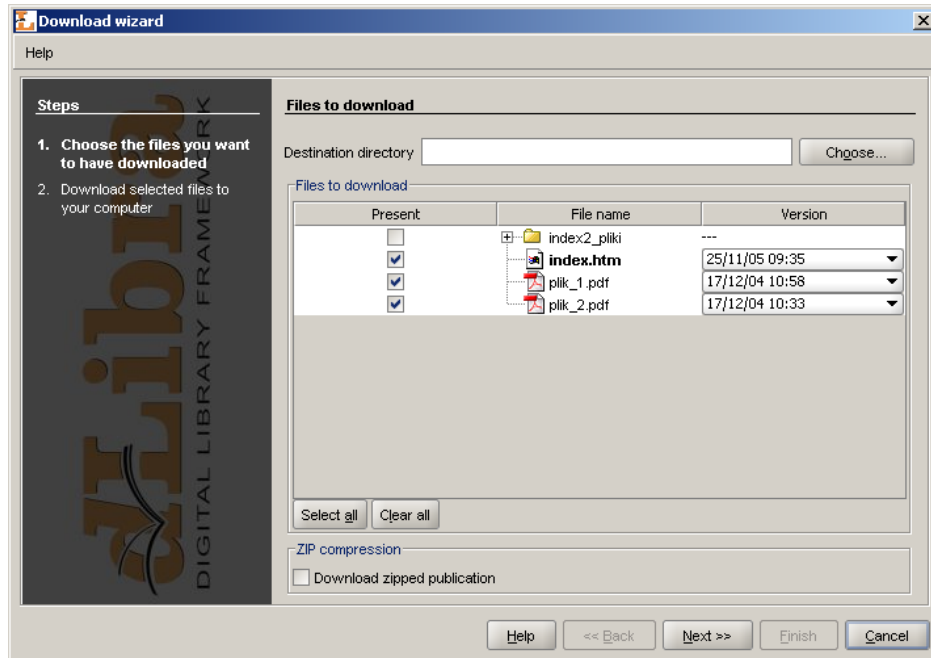
In the Files to download table the default file selection can be altered. To exclude a file from the download list --- uncheck the appropriate position in the Present column. Additionally, the version to be downloaded can be changed using a pull down list in the Version column.

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.

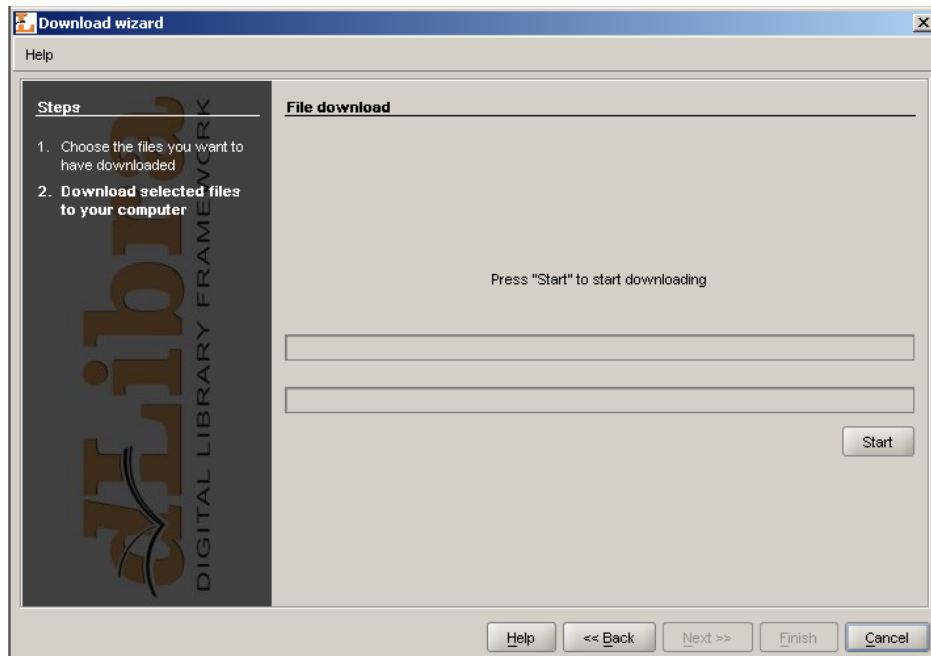


**Figure 3.23. Download Wizard - publication file selection page**



- In the second step (Figure 3.24, "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 3.24. Download wizard - file download page**




### 3.4.3. Adding new files' versions

After modifying some of files of an existing publication, for the changes to take effect the files must be sent back to the server (and the for example new edition can be created based on newly added filed).

To send the updated files to the server:

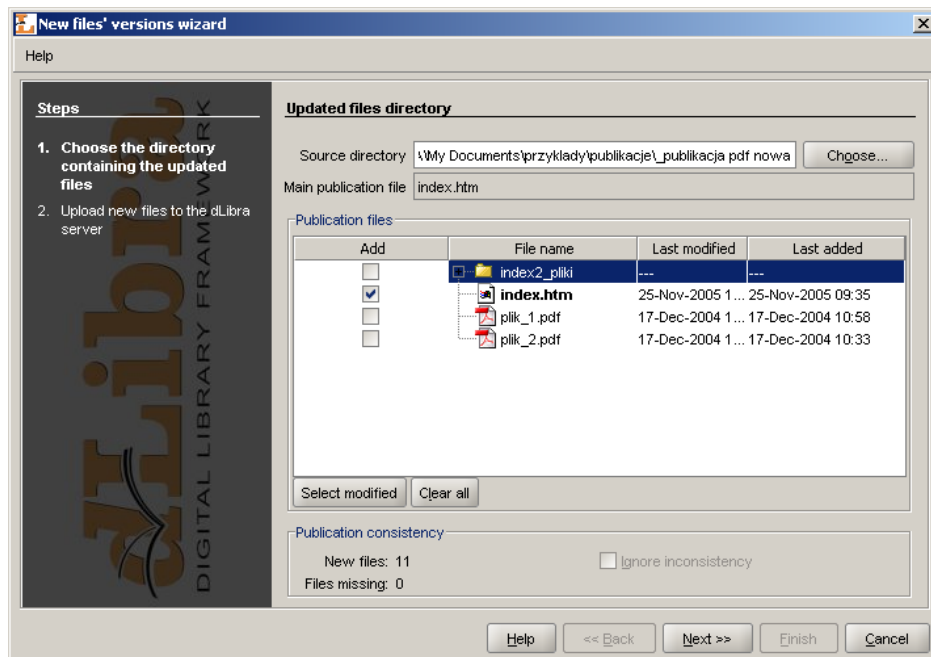
1. On the elements list select the publication to which the new file versions shall be assigned.
2. From the context menu select the Add new files' versions... function. Alternatively, the toolbar  button or main menu options may be used.
3. In the Add new files' versions wizard select the local disk 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.

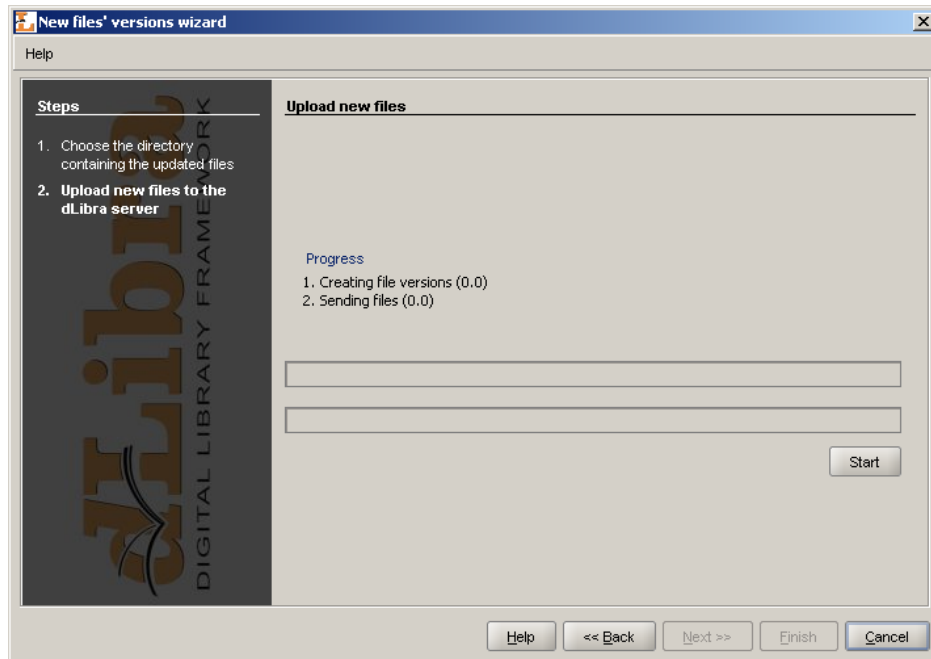
**Figure 3.25. File Update Wizard - updated files directory page**



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

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


**Figure 3.26. File Update Wizard - file upload page**



### 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:

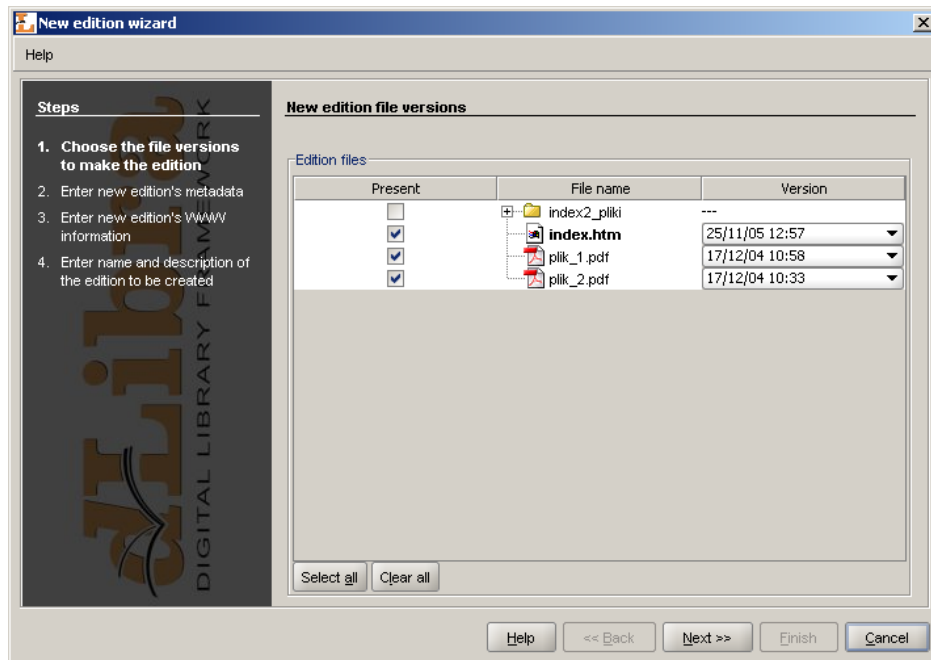
1. On the elements list select the edition (e.g. the current edition) the new edition should be based on.
2. From the context menu select the New edition... function. Alternatively, the toolbar  button or main menu options can be used.
3. In the New Edition Wizard select the versions of publication files 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.

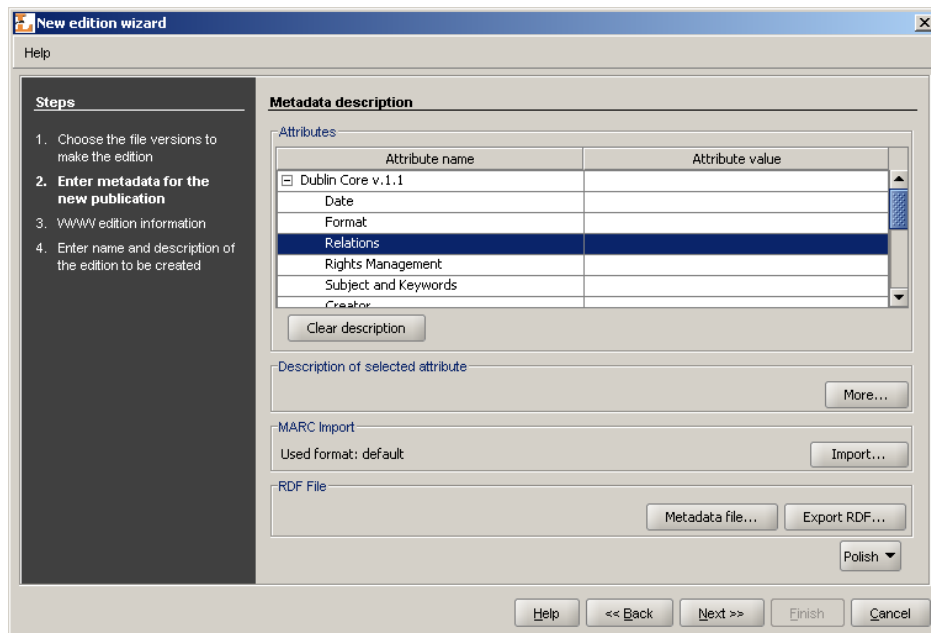
**Figure 3.27. New Edition Wizard - file versions selection page**



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.

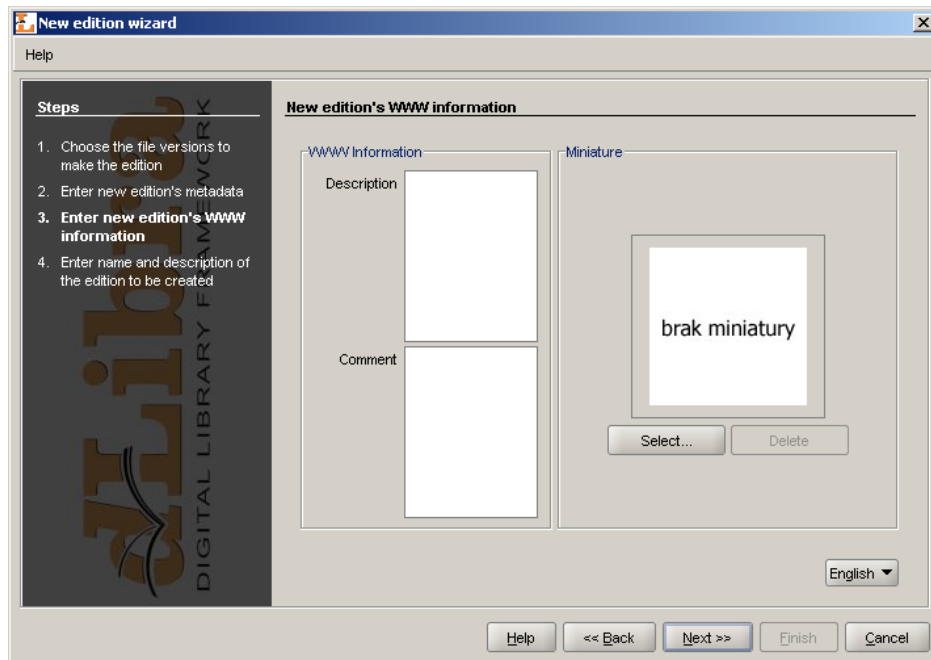
**Figure 3.28. New Edition Wizard - bibliographic description page**



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 3.29. New Edition Wizard - WWW information**



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 3.30. New Edition Wizard - edition properties page**

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 versions upload creator select Mass publication upload from Tools menu.

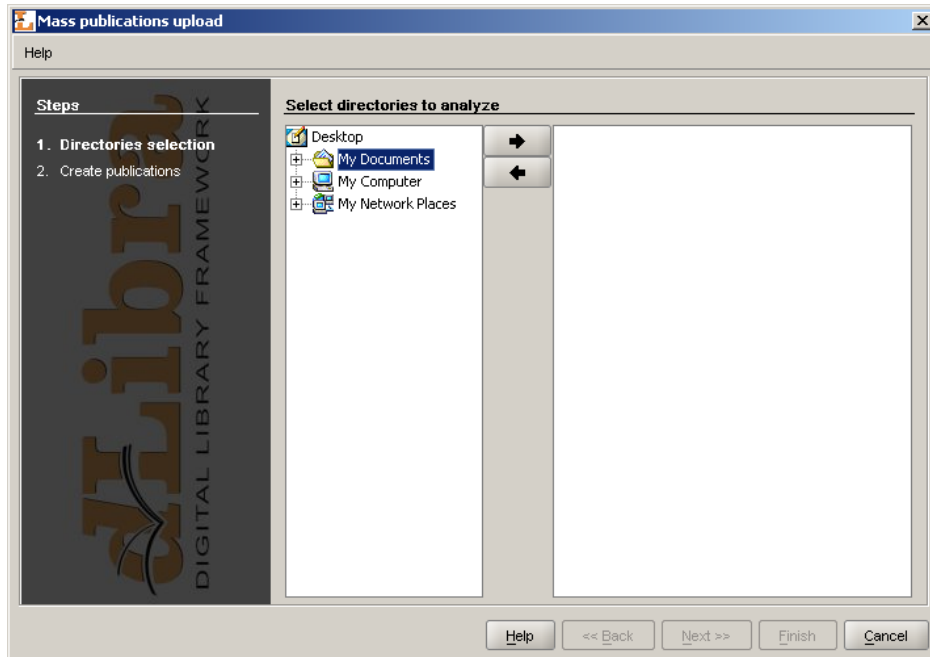
1. In the first step (Figure 3.31, “Mass publication upload creator - publications selection.”) we specify publications which will be sent to *dLibra* server.

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.

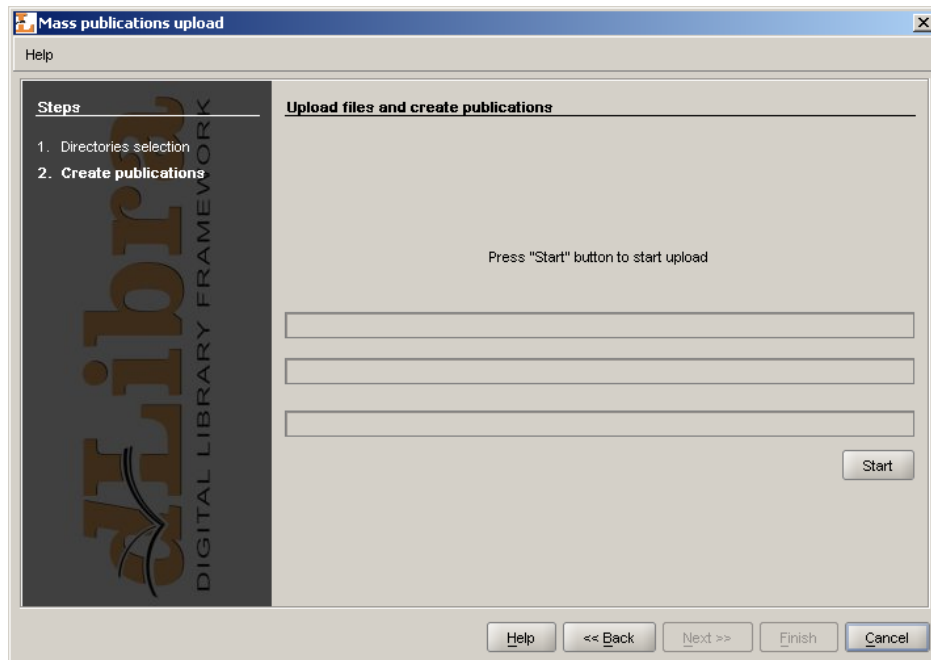
After adding publications to send it is possible to go to the next step.

**Figure 3.31. Mass publication upload creator - publications selection.**



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

Figure 3.32, “Mass publication upload creator - publications upload” shows a window of the second step. To send publications press Start button. During the upload process 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 3.32. Mass publication upload creator - publications upload**

### 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`.

Exemplary `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 exemplary 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 “Exemplary 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. If it is impossible to gain all these informations it is impossible to create 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 recursively. 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`).

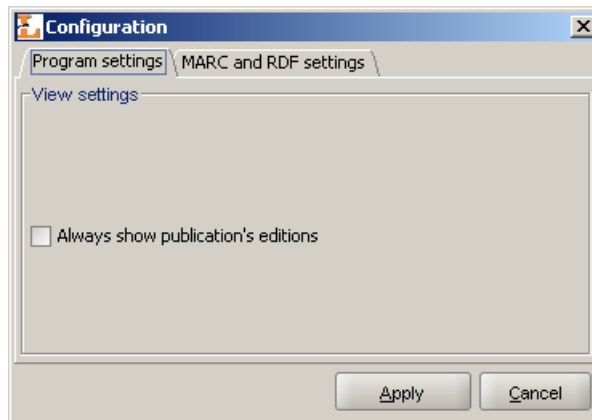
#### **3.4.6.2.4.**

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 3.33, “Application's configuration window”).

**Figure 3.33. Application's configuration window**



Tabs placed in configuration window are responsible for configuration of particular application features. In the following sections all the tabs are described.

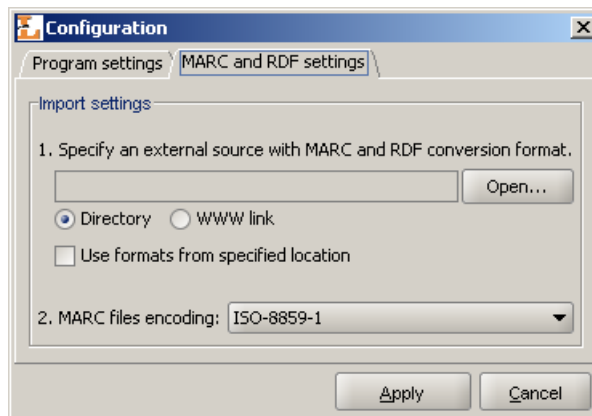
### 3.4.7.1. Program configurarion

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".

### 3.4.7.2. MARC and RDF settings

MARC and RDF settings tab allows adjusting bibliographic description import from files in *MARC 21 communication* and *RDF* format. On the Import settings panel two points are placed.

**Figure 3.34. Konfiguracja - zakładka Ustawienia MARC i RDF**



1. Configures imports from both formats. Here, you can specify source of external configuration. The source may be either directory (Directory option) or web link (WWW link option). In both cases in specified place (directory on hard drive or on the web page) three configuration files should be present:

- `marcImport.properties` file - configuration file for MARC import (???),
- `rdfImport.properties` file - configuration file for RDF import (Appendix B, *Import from and export to RDF format*),
- `namespaces.properties` file - configuration file for RDF import (Appendix B, *Import from and export to RDF format*).

In case of Directory option you have to select directory which contains configuration files by pressing the Open... button. In case of WWW link option you have to type web page address in the text field and press Download button in order to retrieve configuration files.

If the Use formats from specified location check box is selected then the configuration from specified location is used, otherwise default configuration is used (Appendix D, *Editor's application default configuration*).

2. It concerns only MARC configuration. On the MARC files encoding list you can select encoding used to read files in *MARC 21 communication* format.

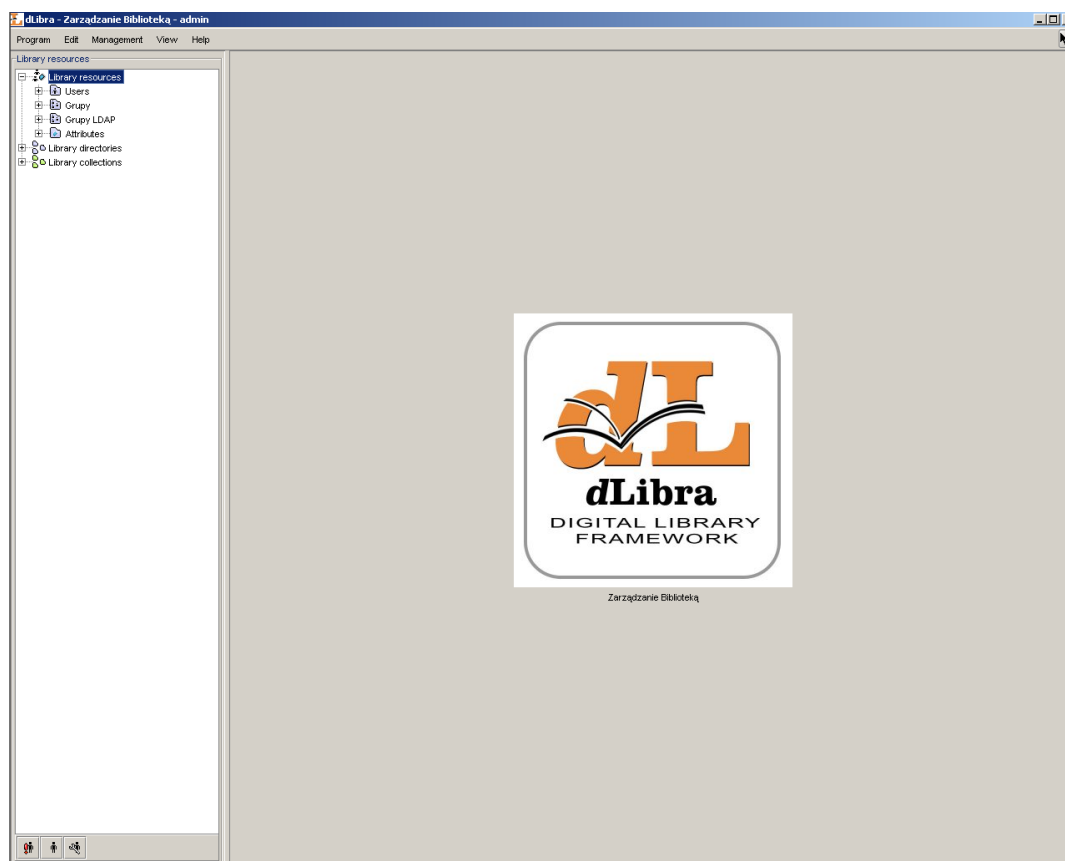
---

# 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. The toolbar provides quick access to the most commonly performed operations (e.g. creating a new publication).


### 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 when *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.

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 have hierarchical structure and generally split library thematically (Section 1.4, “Library collections - reader's view”). A context menu is associated with collection node. It provides quick access to three operations: adding, deleting 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.

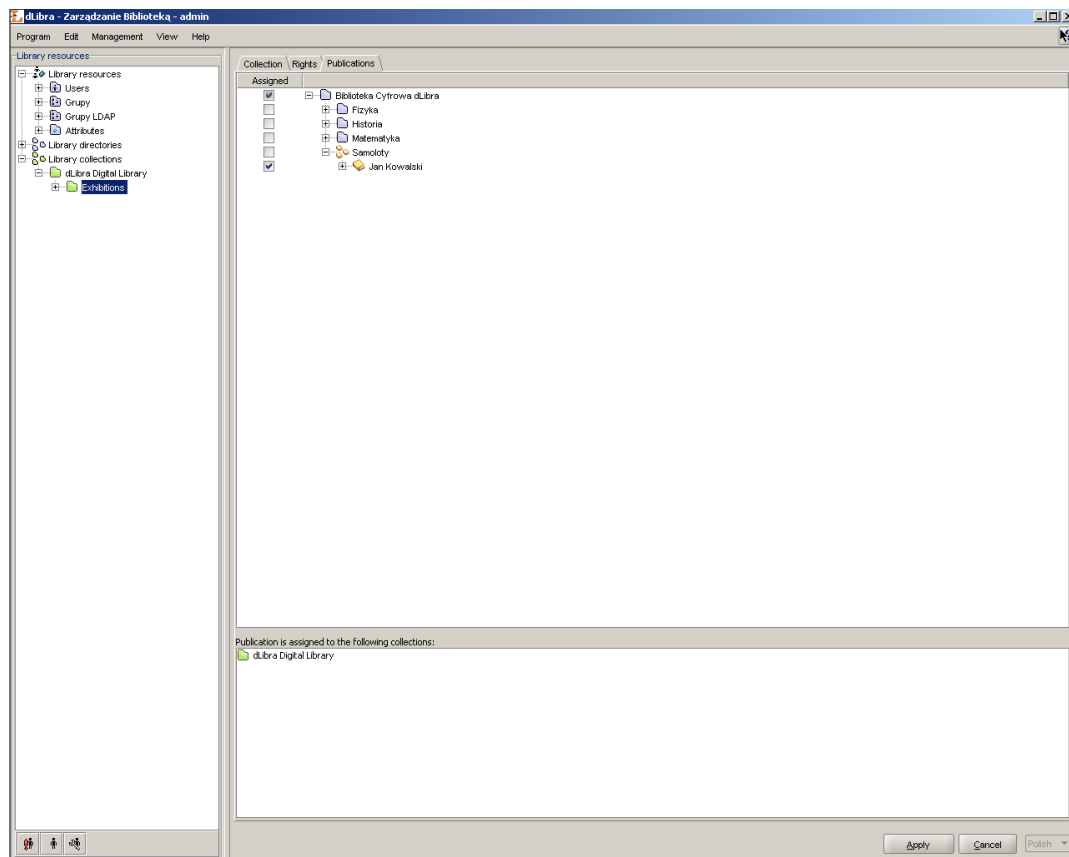
With every node a context menu is associated that provides quick access to the most common operations performed on a given type of element (e.g. creating a new directory).

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

## 4.1.2. Managing publications in 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.

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 4.2. Collection's publications management**

After choosing certain collection in “Collections Tree”, on the tab named Publications (Figure 4.2, “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. Basic administrator's tasks

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

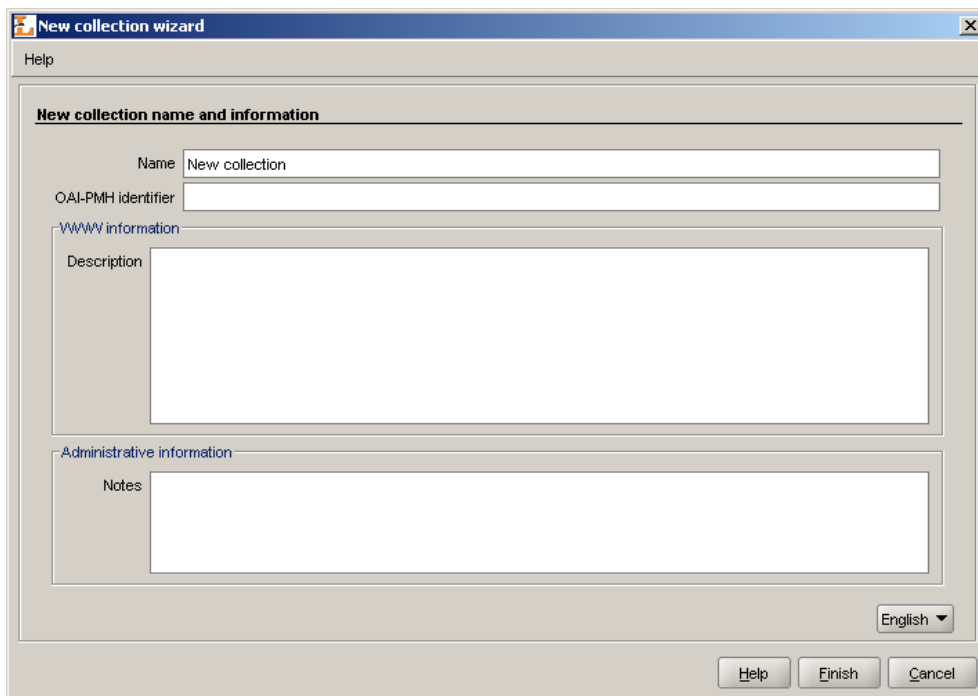
### 4.2.1. Creating/removing collection

#### 4.2.1.1. Creating and removing library collections

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 New Collection Wizard enter name, description and notes for the new collection. Press Finish button to create the collection.


**Figure 4.3. New Collection Wizard**



To remove a collection:

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



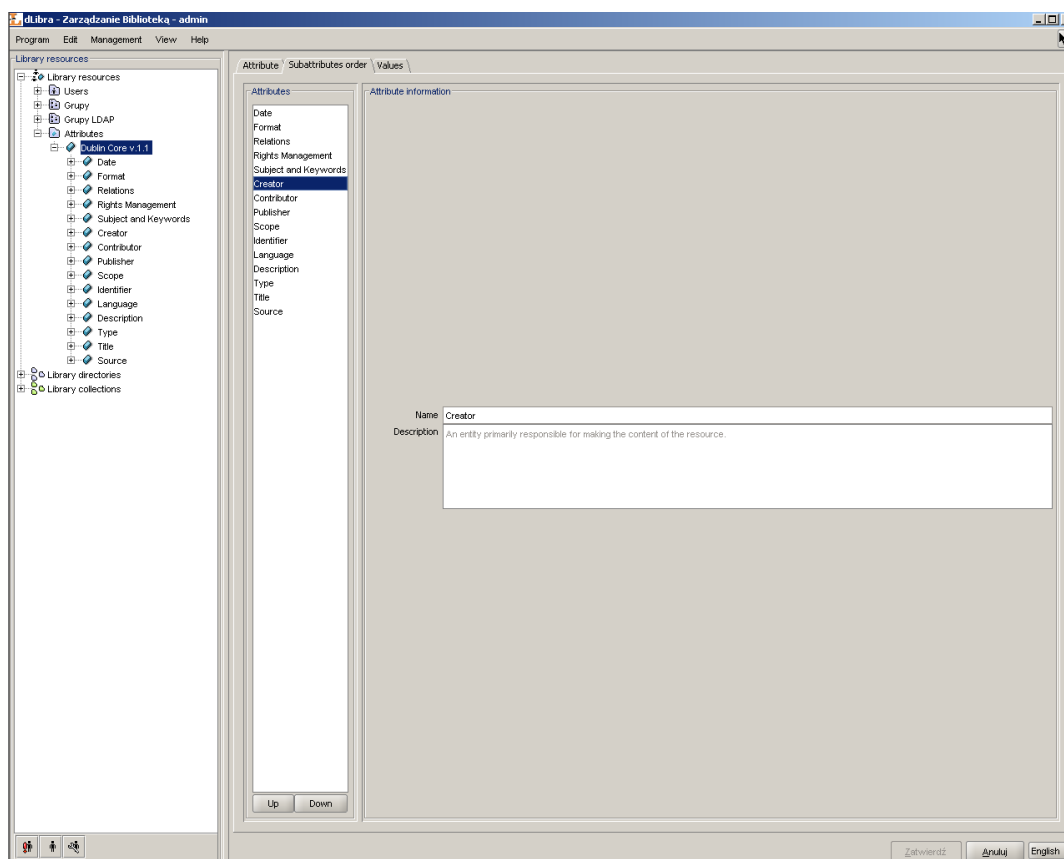
2. Press the  Delete button. Alternatively, a context menu or a pull-down menu can be used.

## 4.2.2. 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 4.4, “Attributes order management”).

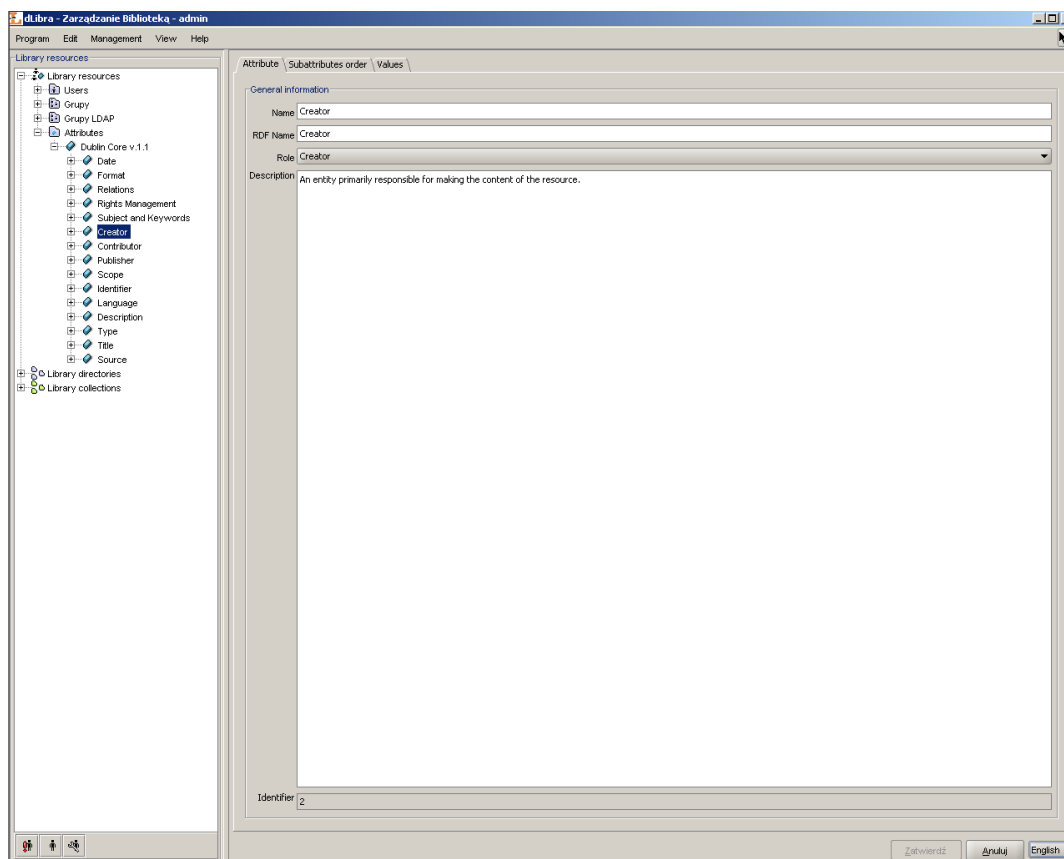
**Figure 4.4. Attributes order management**



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 4.5. Attribute properties



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

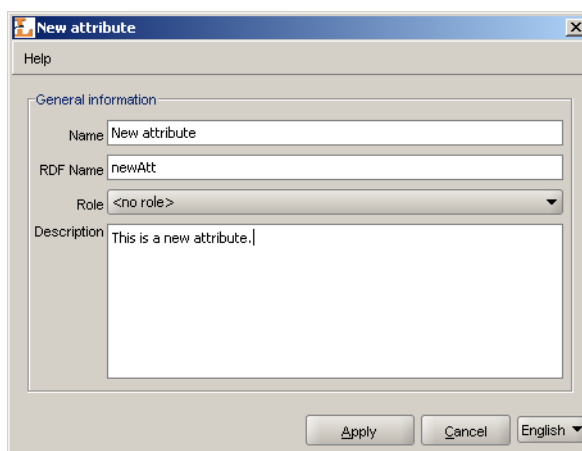
- Resource type
- Rights management
- Source
- Subject and keywords
- Title

### 4.2.2.1. 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 4.6. 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.

### 4.2.2.2. Attributes editing

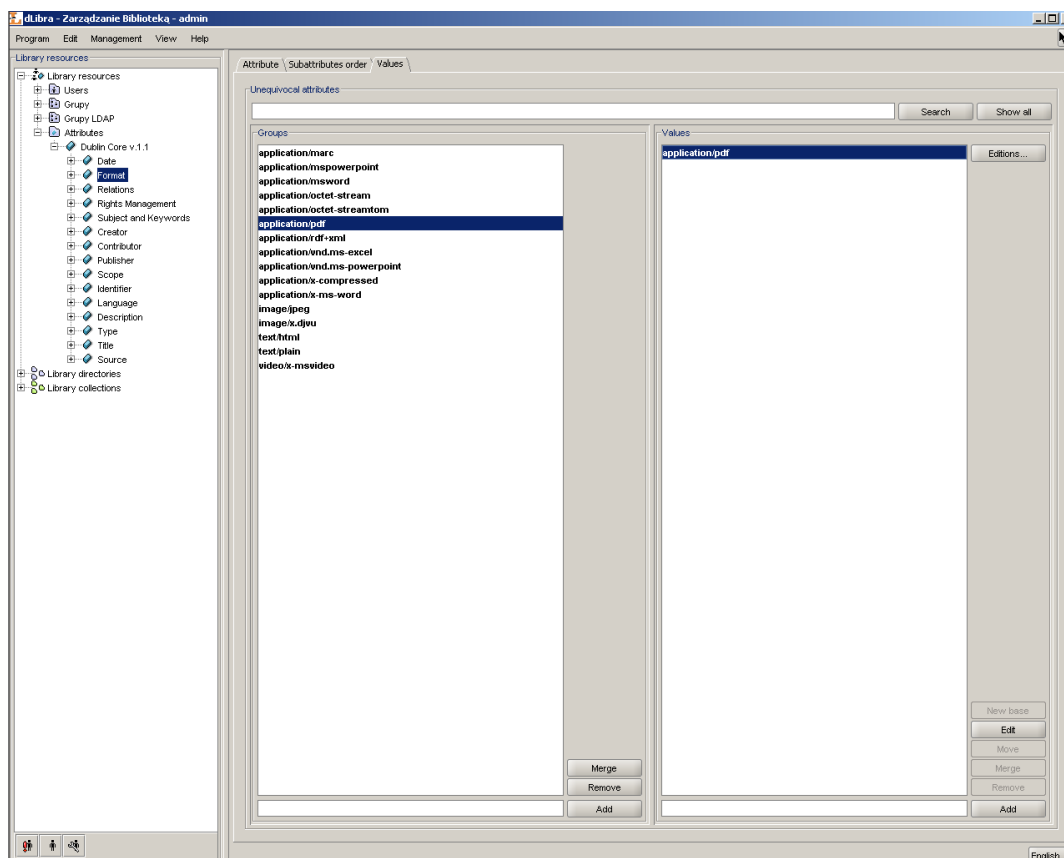
To edit properties of an attribute:

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.

### 4.2.3. 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 4.7. 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 of 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.

### 4.2.3.1. 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:

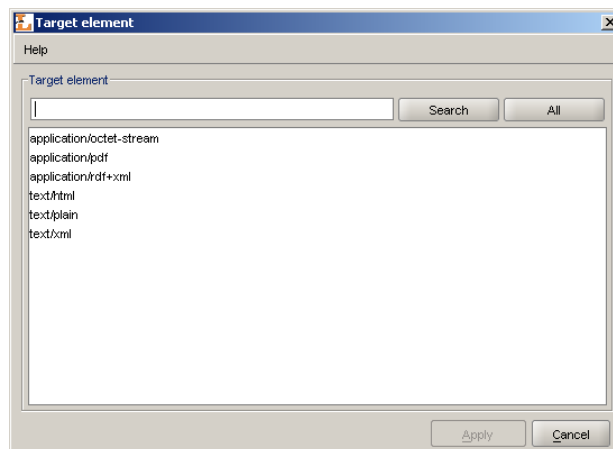
1. Choose a single group on a groups' list.
2. Press Remove button placed beside the groups' list. After confirming, a group and all it's values will be removed from the list and from the system.

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 4.8, "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 4.8. 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.

### 4.2.3.2. 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 4.9. Editing an attribute's value's name**



To merge two values in a group:

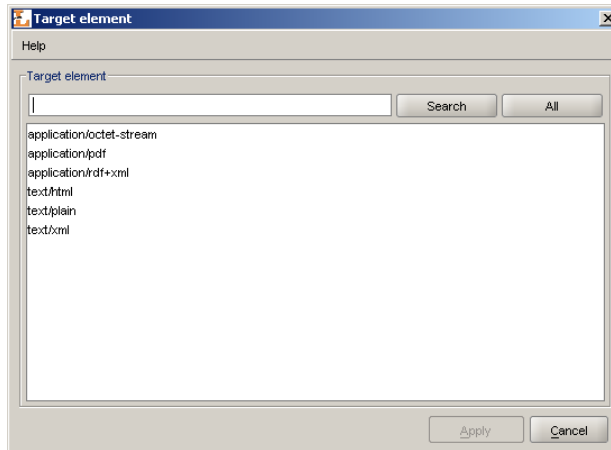
1. Choose the value that should be merged with other value.

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

**Figure 4.10. 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:

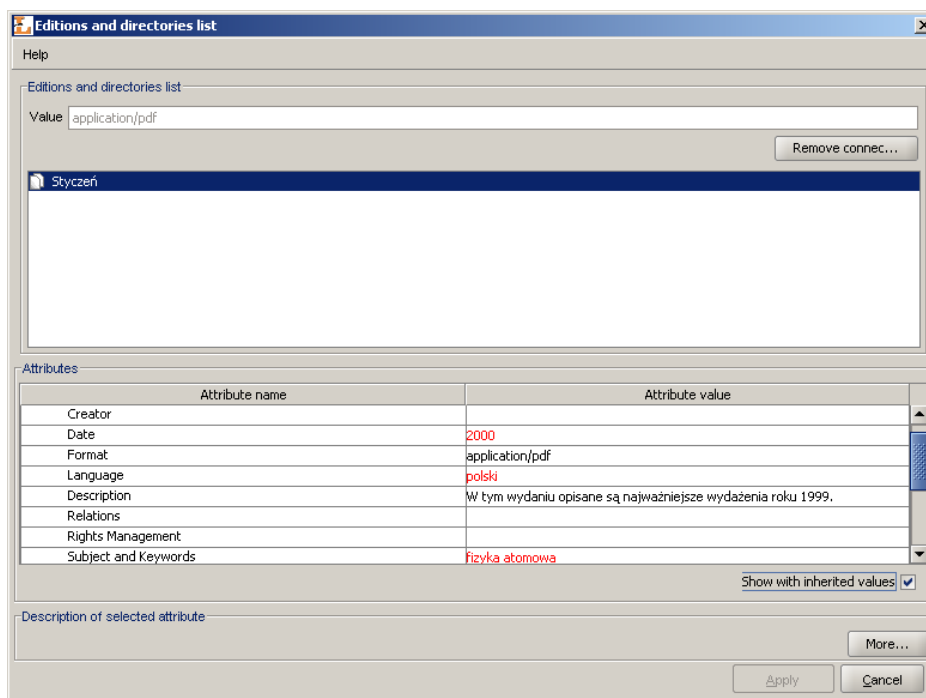
1. Choose a value form the list that has to be the new base value.
2. Press New base button. Chosen value will become a new base value in a group and therefore a group's name will change into a new one.

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 4.11. List of editions and directories connected with a value**



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. Users and groups management

### 4.2.4.1. Library users management

To create a new library user:

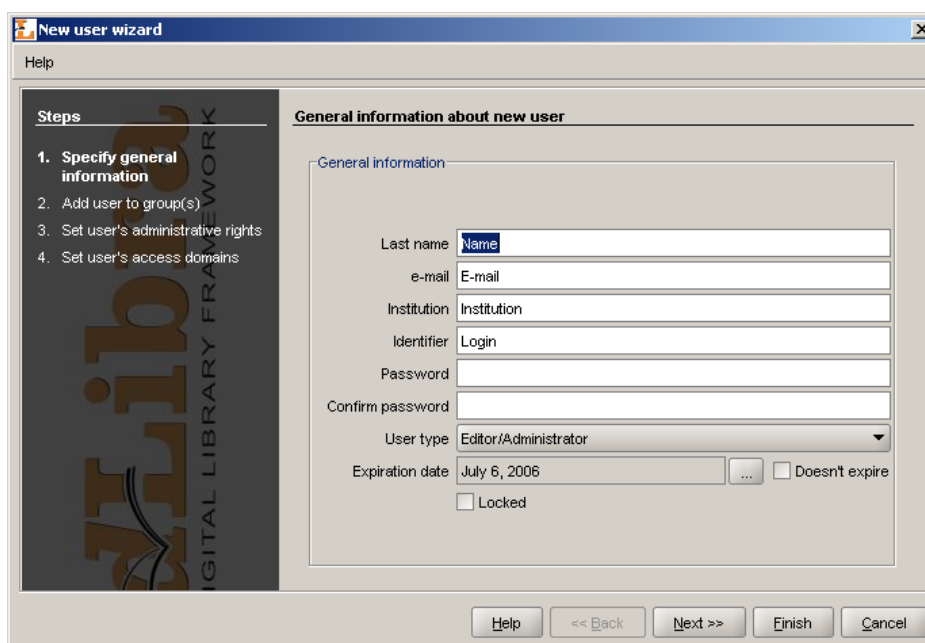
1. In the Library Tree select the users node or a user node.



2. From the popup menu select the New user... option. Alternatively choose  button from toolbar or opcję New user... from Management menu.
3. In the first step (Figure 4.12, “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.

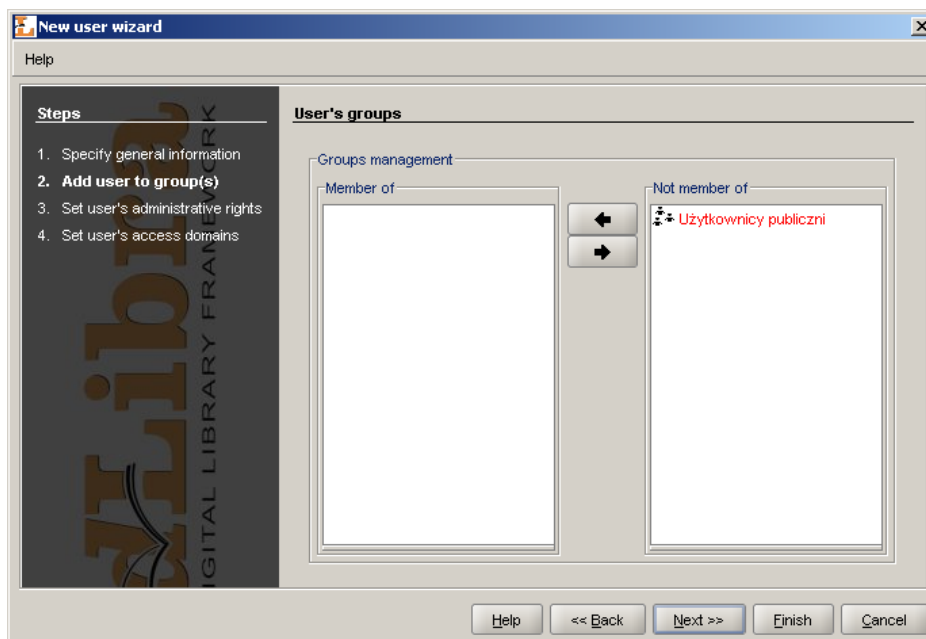
To go to the next step, press the Next button. It is possible to create the user here by pressing the Finish button.

**Figure 4.12. New user wizard - general information**

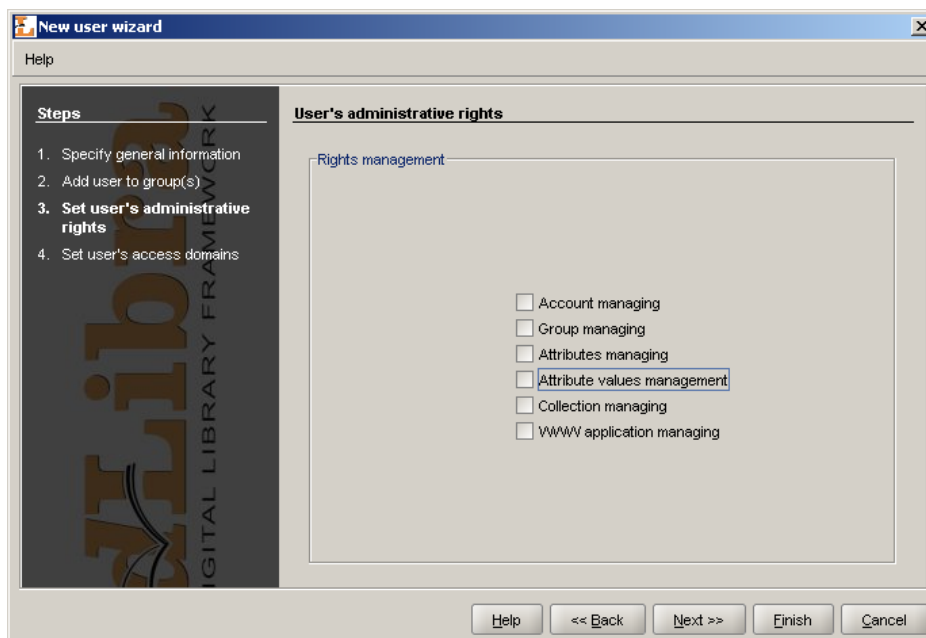


The screenshot shows a window titled "New user wizard" with a "Help" button in the top-left corner. On the left side, there is a "Steps" panel with a vertical list of four steps: 1. Specify general information (highlighted), 2. Add user to group(s), 3. Set user's administrative rights, and 4. Set user's access domains. The main area is titled "General information about new user" and contains a "General information" section with the following fields: "Last name" (with "Name" entered), "e-mail" (with "E-mail" entered), "Institution" (with "Institution" entered), "Identifier" (with "Login" entered), "Password" (empty), "Confirm password" (empty), "User type" (a dropdown menu showing "Editor/Administrator"), "Expiration date" (a date field showing "July 6, 2006" and a "..." button), and a "Doesn't expire" checkbox. Below these fields is a "Locked" checkbox. At the bottom of the window, there are five buttons: "Help", "<< Back", "Next >>", "Finish", and "Cancel".

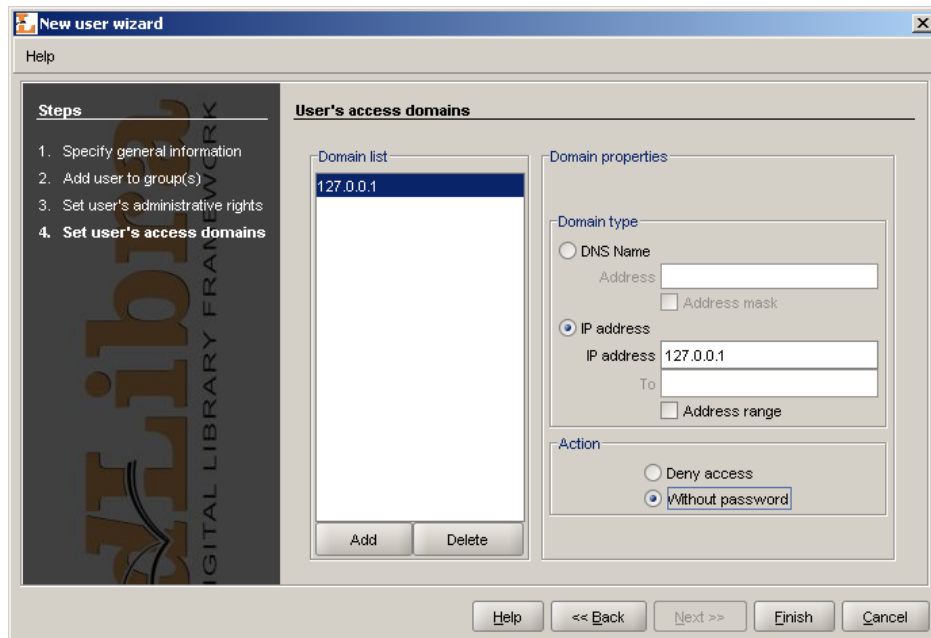
4. In the second step (Figure 4.13, “New user wizard - assigning to groups”) assign user to groups. Member of list contains groups which the user will be assigned to, Not member of list contains groups which the user will not be assigned to. Press the Next button to go to the next step.

**Figure 4.13. New user wizard - assigning to groups**


5. In the third step (Figure 4.14, “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 4.14. New user wizard - assigning administrative rights**

6. In the last fourth step (Figure 4.15, “New user wizard - defininf access domains”), assign access domains. Managing access domains is described in details in ???.


**Figure 4.15. New user wizard - defininf access domains**

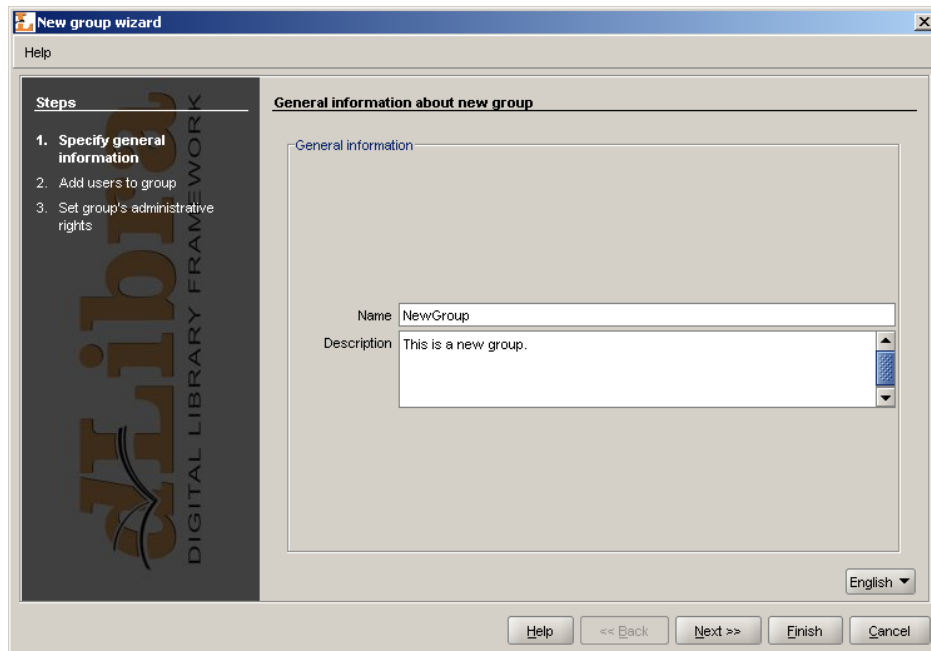
To remove a user:

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.

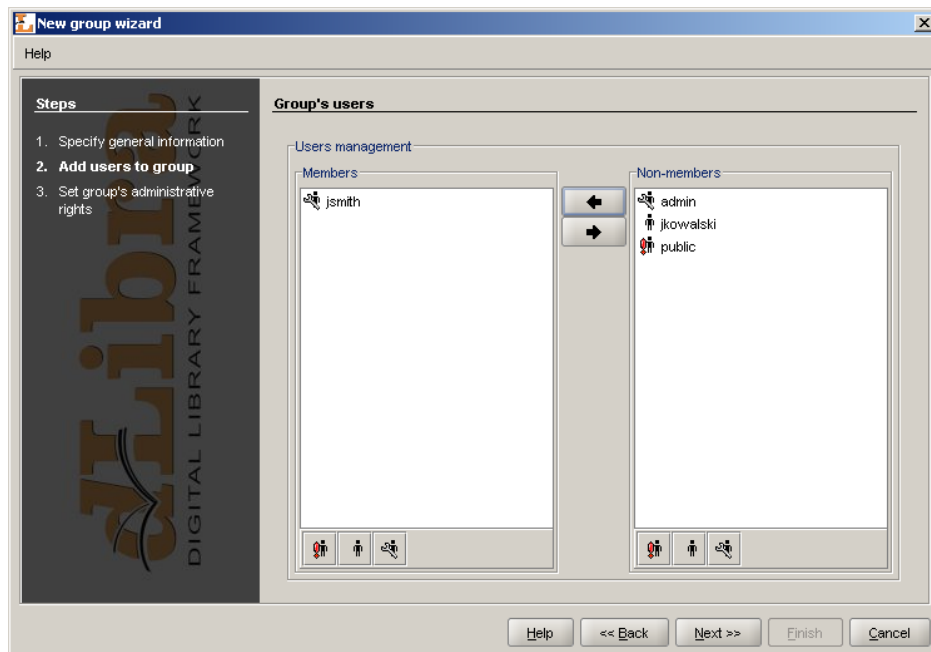
#### 4.2.4.2. Library groups management

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 4.16, "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.

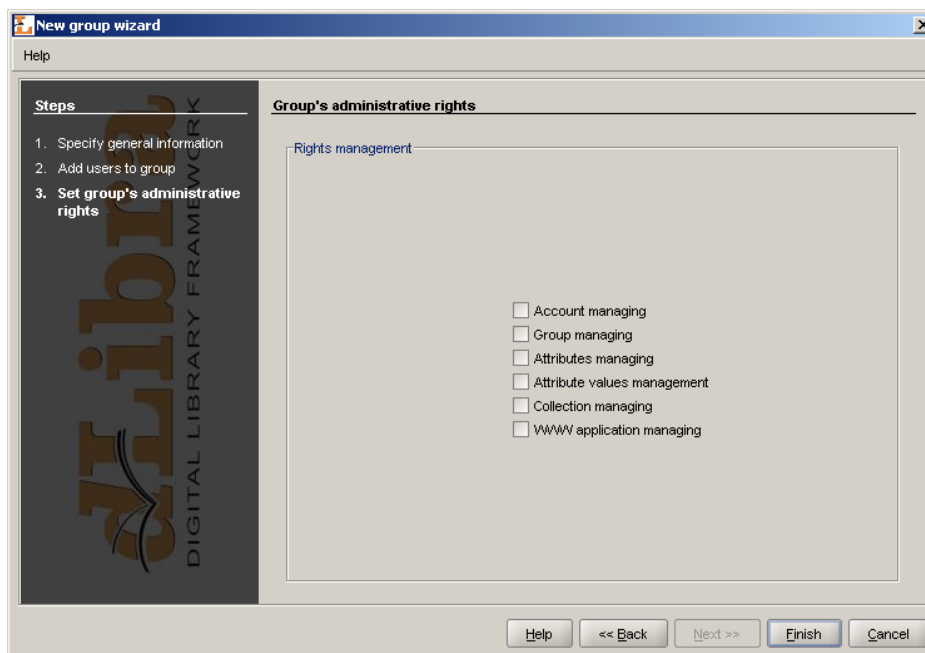
**Figure 4.16. New group wizard - general information**

4. In the second step (Figure 4.17, “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 possible to create the group at this point by pressing the Finish button.


**Figure 4.17. New group wizard - assigning users**

- In the last step (Figure 4.18, “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 4.18. New group wizard - assigning administrative rights**



To remove a group:


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

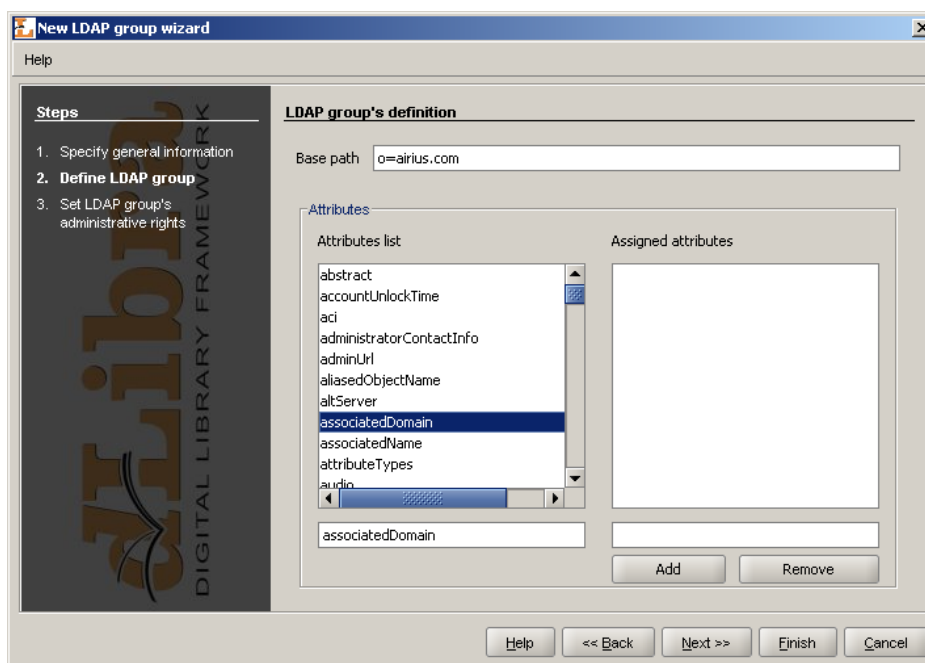
### 4.2.4.3. LDAP groups management

#### Note

To understand this section basic knowledge about LDAP is required.

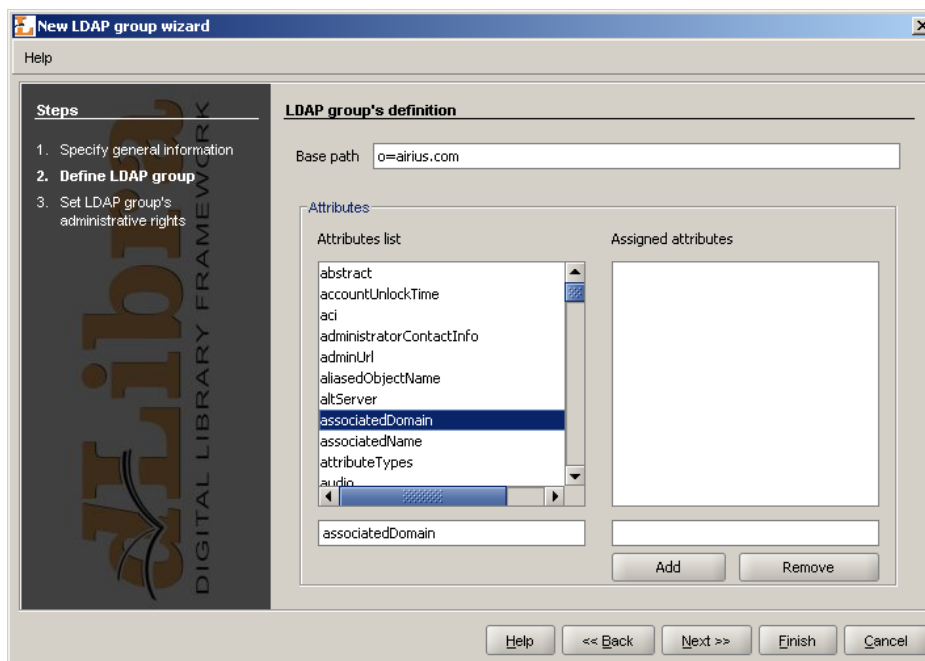
To create a new LDAP group:

- In the Library Tree select the LDAP groups node or an LDAP group node.
- Select the  button from toolbar or choose New LDAP group... from popup menu. You can also choose New LDAP group... option from Management menu.
- In the first step of new LDAP group wizard (Figure 4.19, “New LDAP group wizard - general information”) specify general information.

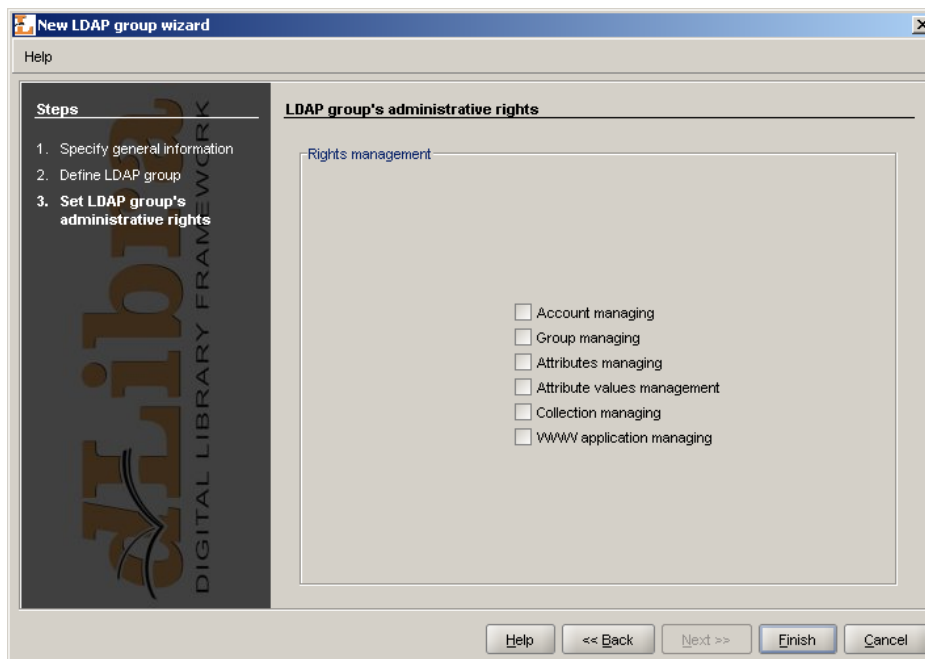
**Figure 4.19. New LDAP group wizard - general information**

4. In the second step (Figure 4.20, “New LDAP group wizard - defining LDAP group”) define group. Assign group attributes 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 possible to create the LDAP group at this point by pressing the Finish button.


**Figure 4.20. New LDAP group wizard - defining LDAP group**

5. In the last step (Figure 4.21, “New LDAP group wizard - assigning administrative rights”) assign administrative right. To create new LDAP group press Finish button.

**Figure 4.21. New LDAP group wizard - assigning administrative rights**

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.

## 4.3. 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

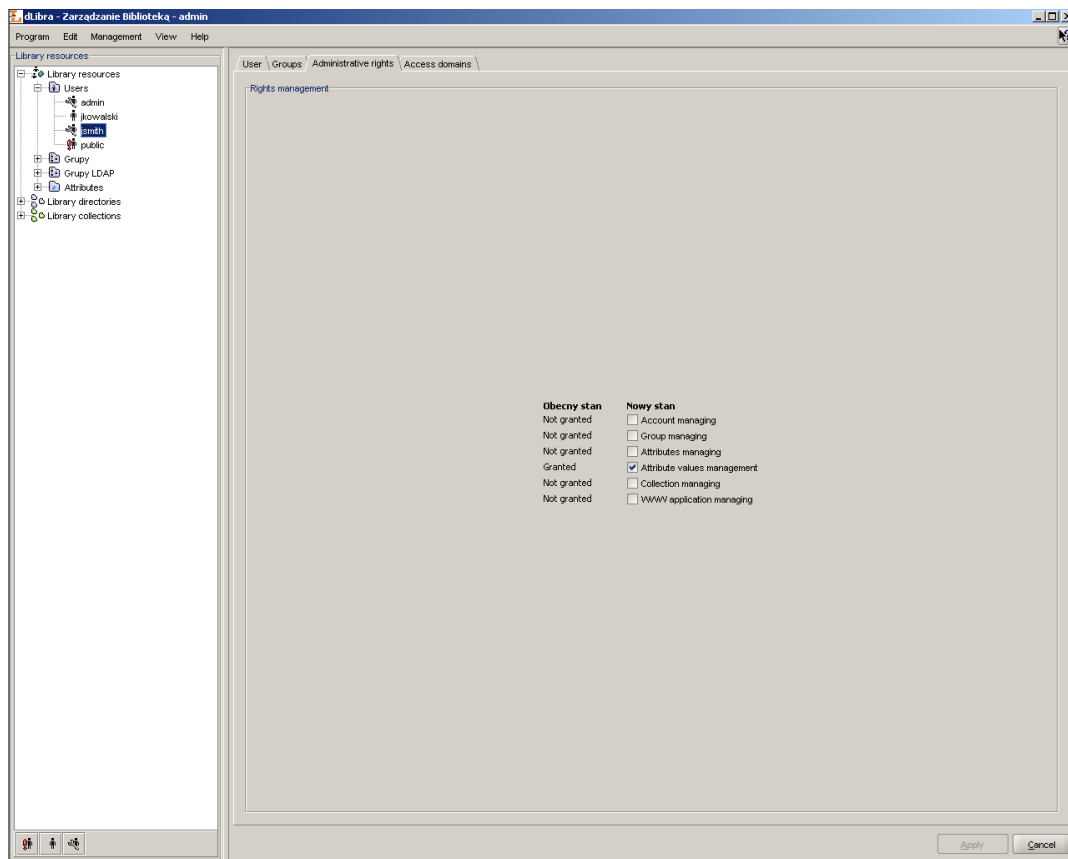
In *dLibra* there are five library-level (administrative) permissions:

- 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
- collections management - allows to create, remove and alter collections

The administrative rights of a user can be changed in the User Properties panel, Administrative rights tab (Figure 4.22, “User administrative rights tab”). The administrative rights of a group can be changed in the Group Properties panel, Administrative rights tab.



Figure 4.22. User administrative rights tab



### 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

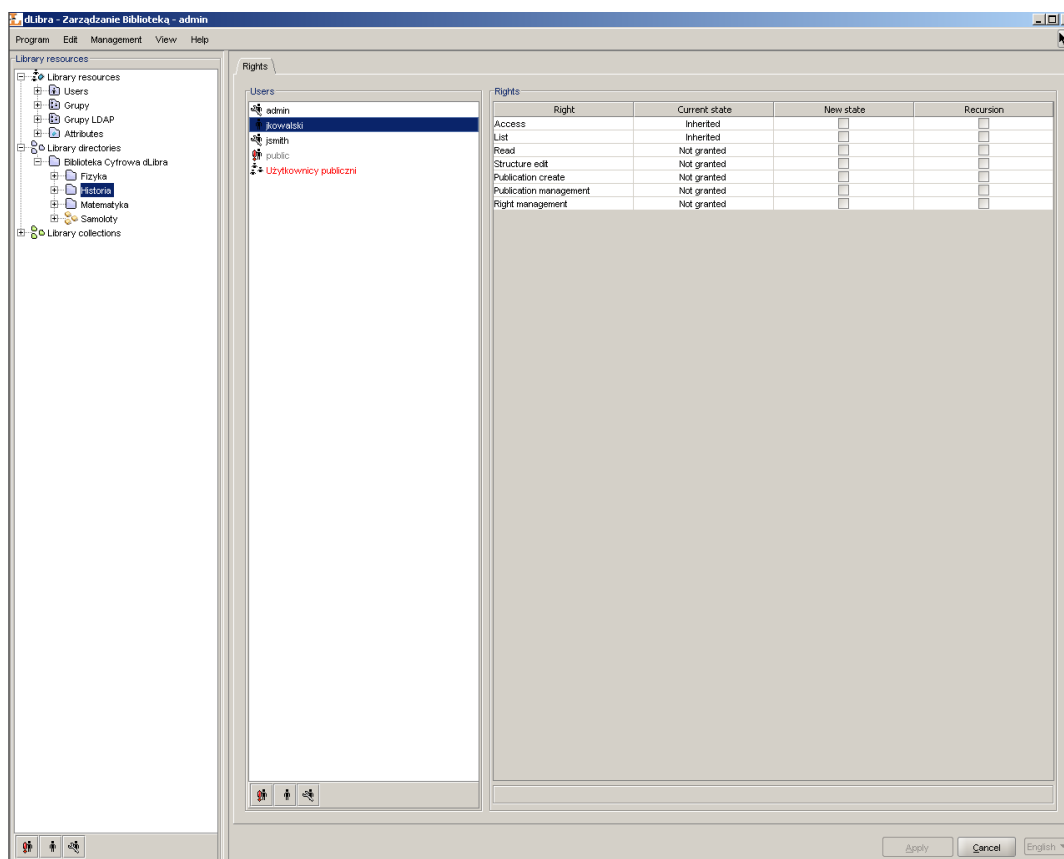
In *dLibra* there are seven directory-level permissions:

- 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 sub-directories) 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 4.23. Directory-level rights editor**



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.

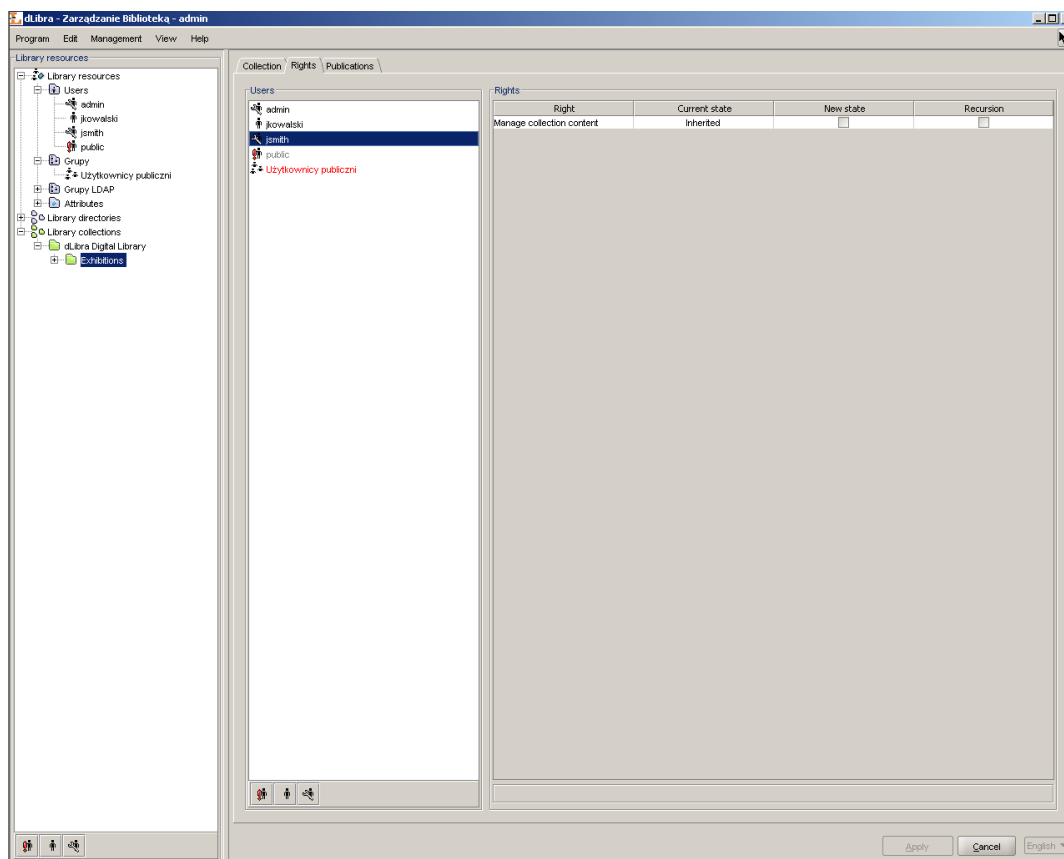
### **4.3.4. Collection-level permissions**

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 4.24. Collection-level rights editor



Rights can be managed on a tab signed Rights.

### 4.3.5. Publication-level permissions

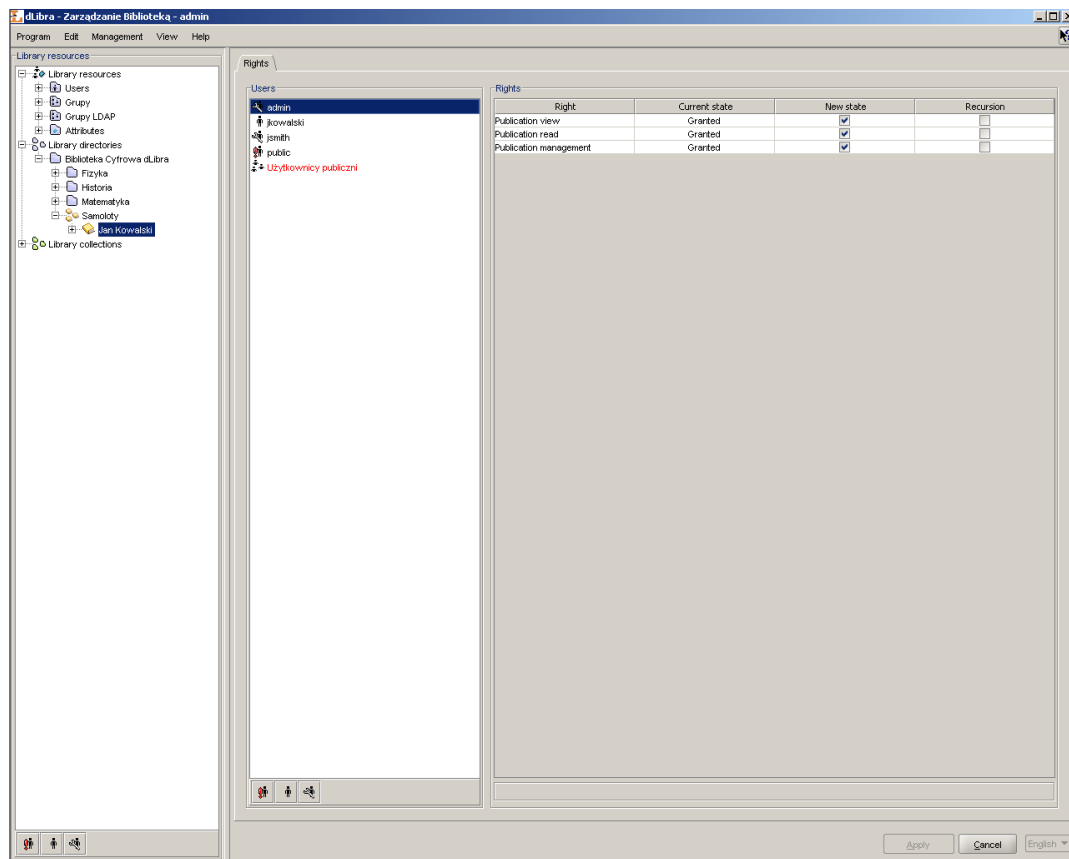
As with the library directories or collections, access to *dLibra* publications is controlled separately for every single publication and every single library user - for the same publication different users can have different permissions, and one user can have different levels of access to different 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 4.25. Publication-level rights editor



## 4.4. Access domains management

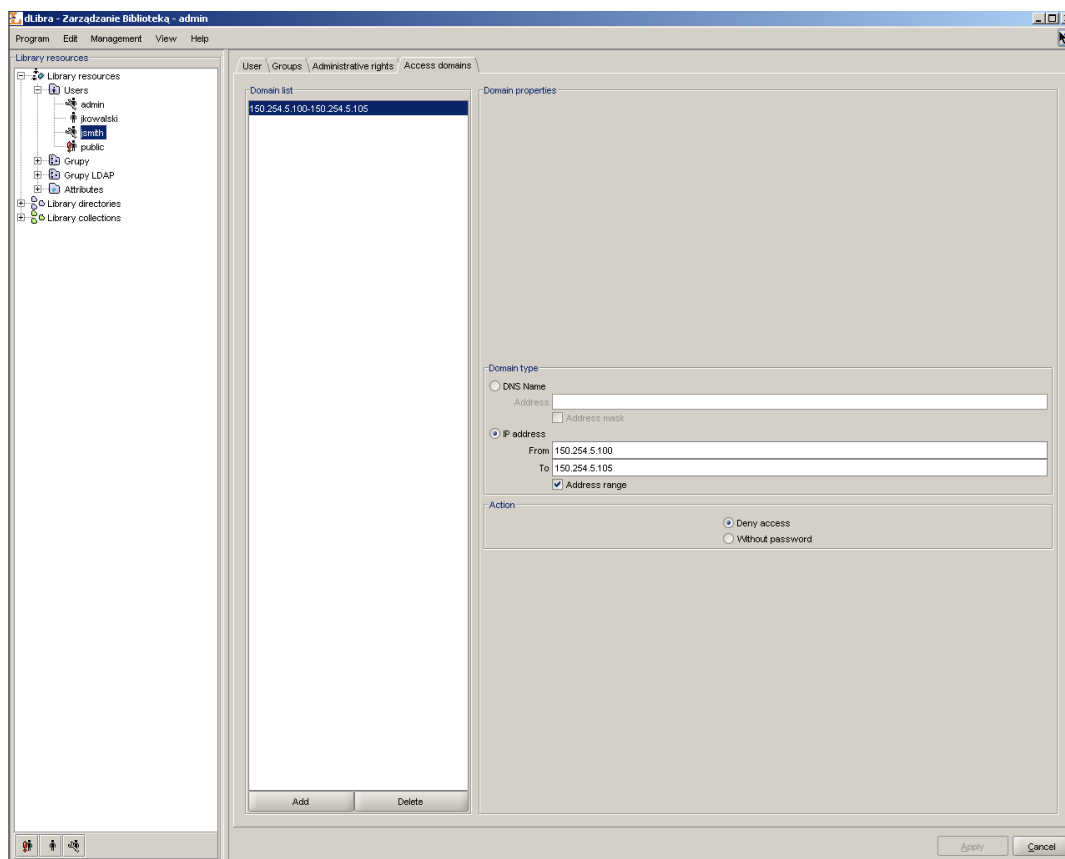
In *dLibra*, for every library user it is possible to define a number of access domains so that, from some places in the Internet, the user can access the library without authentication, from other - access is denied or requires entering a password.

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 4.26. User access domains



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. MARC Import

Attributes' values import from MARC format in *dLibra* application is based on default configuration built-in inside the application. This configuration is unchangeable though it is possible to use the other configuration defined in text files outer to the application (see Section 3.4.7, "Editor's application configuration"). The files have a simple properties file format. This section describes this format and shows how to create these files to change a default application import configuration.

## A.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;
```

Above there is shown an exemplary content of the MARC configuration file.

Each line in configuration file consists of attribute and its value which are divided with equation sign. Depending on which attributes have to be imported from MARC document there have to be RDF names of these attributes in separate lines. After equation sign there is kept an information about which MARC fields are to be written to certain attribute's value. It consists of field numbers written in a particular way described later in this section. All field numbers have to be written in the same line without any separation character. The lack of any attribute name means the same as leaving an empty set of field numbers by this attribute. In this situation no values will be imported to such an attribute.

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 shown 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 another 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 characters 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 range 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 an 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**.



## A.2. marcImpRemChars.properties configuration file format

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

An exemplary `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.

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). `begin` or `end` separates the place 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 cause that the application will remove the `ab` characters sequence from the beginning of 245b subfield, if this sequence will be found at the beginning 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.

---

# Appendix B. Import from and export to RDF format

## Note

This appendix described configuration files used for import and export bibliographic description. 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.

## B.1. RDF configuration files

There are two *RDF format* configuration files:

- `rdfImport.properties` - configuration file which is used when importing metadata from RDF files to *dLibra* attributes' values.
- `namespaces.properties` - namespaces configuration file - namespaces are commonly used in the XML technology which RDF is based on.

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

```
wlasnosc=wartosc wlasnosci  
wlasnosc2=wartosc wlasnosci 2
```

### B.1.1. namespaces.properties file

`namespaces.properties` is used for:

- shortcut specification for namespaces. These shortcuts are used in `rdfImport.properties` file.
- *dLibra*. namespace specification which is used when importing and exporting metadata. The namespace is the value of `systemURL` key.

Exemplary `namespace.properties` file:

```
systemURL=http://dlibra.psnc.pl/  
dc=http://purl.org/dc/elements/1.1/  
wbc=http://www.wbc.poznan.pl/
```

In this file library namespace is defined as “`http://dlibra.psnc.pl/`”, the shortcut for Dublin Core 1.1 (`http://purl.org/dc/elements/1.1/`) schema is “`dc`”, and the shortcut for Wielkopolska Biblioteka Cyfrowa (`http://www.wbc.poznan.pl/`) is “`wbc`”. Library namespace will be used for export to RDF, for example:

```
<?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>
```

Above file describes publication in the library namespace which is “http://dlibra.psnc.pl/” (*systemURL* key in the *namespaces.properties* file).

## B.1.2. *rdfImport.properties* file

In the *rdfImport.properties* file we define which attributes from imported namespace correspond to which attributes in the *dLibra* system. Exemplary import file:

```
Title=dc:title;wbc:Title ❶
Access=dc:rights;wbc:Rights ❷
MimeType=wbc:Format ❸
```

- ❶ This line means that to attribute which has “Title” *rdName* values from “title” of “dc” schema and “Title” of “wbc” schema will be added.
- ❷ This line means that to attribute which has “Access” *rdName* values from “rights” of “dc” schema and “Rights” of “wbc” will be added.
- ❸ This line means that to attribute which has “MimeType” *rdName* value from “Format” of “wbc” schema will be added.

Please notice that in *rdfImport.properties* file shortcuts defined in *namespaces.properties* file are used.

---

# Appendix C. Attributes' values' Synonyms Dictionary

## C.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.

## C.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

Attributes' values' Synonyms  
Dictionary

---

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 D. Editor's application default configuration

## D.1. Default configuration files

Three configuration files configure fully editor's application:

- `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.

- `namespaces.properties`:

```
systemURL=http://www.dlibra.psnk.pl/
### EXTERNAL NAMESPACES ###
dc=http://purl.org/dc/elements/1.1/
```

- `rdfImport.properties`:

```
Title=dc:title
Creator=dc:creator
Subject=dc:subject
Description=dc:description
Publisher=dc:publisher
Contributor=dc:contributor
Date=dc:date
Type=dc:type
Identifier=dc:identifier
Source=dc:source
Language=dc:language
Relation=dc:relation
Coverage=dc:coverage
Rights=dc:rights
```

## **D.2. Default editor's application settings**

Editor's application by default is set to simple mode (Section 3.4.1, "Editor's application modes (views)"). MARC and RDF configuration is set to use default internal files.

---

# 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 communication 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   |