

# [EN] 02. The Structure of a Directory which Describes a Publication

## Basic Information

A directory which describes a publication consists of:

- a `publication.properties` file which describes the publication,
- a file with the bibliographic description of the publication (in other words, a metadata file), and
- the files comprising the publication.

## The `publication.properties` file

The `publication.properties` file is the publication properties file. It consists of key-value pairs. A key, that is, the name of a property, is separated from the value with the equal sign (=). Each such pair is in a separate row. The file should be saved as UTF-8 encoded.

The permitted properties and their values are presented below.

- The `publication.actorsRights.<the login of the user or the name of the group to which the permissions are assigned>` key.  
Comma-separated permissions. Note: if the login contains spaces, they must be preceded with the “\” character.  
The list of possible permissions (for more information about permissions, see [here](#)):
  - `pv` – access only to published publication versions,
  - `pr` – access to all publication editions, and
  - `pe` – publication management.
- The `publication.collections` key  
Comma-separated identifiers of collections to which the given publication is to be assigned.
- The `publication.destination.directoryId` key  
The identifier of the directory in which the given publication is to be put.
- The `publication.destination.parentPublicationId` – the identifier of the group publication in which the publication is to be put.
- The `publication.mainFile`  
The name of the main file of a publication (with the extension). The main file must be in the directory which describes the publication.
- The `publication.metadataFile`  
The name of the metadata file (with the extension). A metadata file must be in the directory which describes the given publication and must be in one of the formats which are supported by the currently installed metadata import extension (usually, those are the RDF, MARC, or Bibtex formats – see the [Importing and Exporting Metadata](#) chapter). If the value for that key is not defined, then the default search is for files with the same name as the name of the main file, with the default extensions for the supported format (for example, `rdf`, `.mrc`, or `.bib`).
- The `publication.name` key  
This is a sequence of characters which determines the name of the given publication. If the value for that key is not defined, the application will try to generate the name on the basis of the attribute values (just like in the [second step of the new publication creator](#)).
- The `publication.notes` key  
This is a sequence of characters which determines the administrative notes for the given publication.
- The `publication.published` key  
This is a sequence of characters which determines if the first edition of the publication is to be published. If the value is `true`, the edition will be published. In any other case, the edition will not be published. The default value is `false`.
- The `publication.secured` key  
This is a sequence of characters which determines if the given publication is to be protected. If the value is `true`, the publication will be protected. In any other case, the publication will not be protected. The default value is `false`.
- The `image.content` key  
The name of the file which contains the thumbnail of the given edition.
- The `publication.publishingDate` key  
The planned date of making the given publication available.

A sample `publication.properties` file:

#### publication.properties

```
publication.actorsRights.publicGroup=pr
publication.collections=2,7,12
publication.destination.directoryId=1
publication.mainFile=index.htm
publication.metadataFile=metadata.rdf
publication.name=Sample Publication
publication.notes=Created by Jan Kowalski
publication.published=true
```

In this example, the `publicGroup` will have the permission to review all publications (the assigned `pr` permission implies the `pv` permission). The publication will be assigned to collections with identifiers 2, 7, and 12. The publication will be put in the directory with identifier 1. The main file of the publication will be the `index.htm` file, and the metadata will be in the `metadata.rdf` file. The publication will be named `Sample Publication`, and the administrative notes – *Created by Jan Kowalski*. The first edition of the newly created publication will be published, and the publication will not be protected.

## Group Publications

In order to make a newly created publication a group publication, the main file of the publication (the `publication.mainFile` key) **cannot** be defined. If the directory which describes the publication contains subdirectories, they will also be analyzed, and the publications described in them will be added as the subordinate directories of the group publication. The subordinate publications can also be group publications (if no main file is defined for them), so the user can create a hierarchy with any number of levels.

Note: if the added publication is not a group publication, then the main file must be determined.

## Planned Publications

The created publication can be a planned publication. For that to be the case, the publication directory can **only** contain two files: `publication.properties` and `description.rdf`. Additionally, file `publication.properties` can contain the `publication.publishingDatekey`, the value of which should be the planned date of publication. The allowed date format is: `yyyy-MM-dd`, where `yyyy` means the year, `MM` – month, and `dd` – day.

## Defining the Location of a Publication

In order to create a publication, the user must decide where to put it in the directory structure and the structure of the group publications of the dLibra system. That can be done with the use of property `publication.destination.directoryId` (the target directory) or `publication.destination.parentPublicationId` (the superordinate group publication). The priority of the second property is higher, so if both are defined, the target directory will be ignored (the publication will be added to the directory which contains the superordinate group publication).

When a group publication is added, the subordinate publications added with it do not require a direct definition of the target location. Since the superordinate group publication is known at once, the `publication.destination.directoryId` and `publication.destination.parentPublicationId` will be ignored.

## Defining the Name of a Publication

An added publication must have a name. The name can be specified in one of two ways:

- directly – specifying the `publication.name` property, or
- indirectly – if that property is not specified, the metadata file will be checked with the view to composing the name of the publication in the same way as in the [second step of the publication creator](#). In such a case, then, the user should take care that the metadata used for composing the name are defined.

## Inheriting properties in `publication.properties` files

The inheritance mechanism has been introduced to facilitate defining the properties of an object when many added publications have the same values. The properties do not have to be defined directly in every directory which describes a publication – they can be placed in the `publication.properties` file in the parent directory (indicated in the [mass object upload creator](#)). They are inherited by all added publications. In particular publications, the user can define the selected properties again – in such a case, the inherited values will be overwritten.

Properties are also inherited in the hierarchy of embedded group publications. Thus, if a group publication is created, the properties of all subordinate publications created with it are inherited from that group publication – unless they are overwritten. However, there are exceptions to that rule:

- properties which define the target location (`publication.destination.directoryId` and `publication.destination.parentPublicationId` – the parent publication and directory result from the directory structure;
- collection identifiers (`publication.collections`) – the described mechanism omits them because the publication [grouping mechanism](#) ensures the inheritance of those properties; and
- permissions for accessing a publication (`publication.collections`) – see above.

## Inheriting Metadata

That mechanism is similar to the mechanism of [inheriting](#) `publication.properties` files, except that the metadata file in the parent directory should be named `publication.rdf`. Thus, if the directory which contains directories which describe publications contains a `publication.rdf` file, the metadata from that file will be ascribed to all the created publications. That inheritance mechanism only operates on the first level of directories – it does not pertain to publications grouped in newly created publications. Metadata from group publications are inherited through the [publication grouping mechanism](#).

## Generating a Sample Property File

With the use of the “Save property file” option from the “Tools” menu, the user can save the information about a publication selected on the element list to a file. The file can become a model for files which should be placed in directories which describe publications during [mass object upload](#).