

# [EN] 04. The Structure of a Directory which Describes a Multi-Format Object Basic Information

## Basic information

A directory which describes an object consists of:

- a `publication.properties` file which describes the object,
- a file with the bibliographic description of the publication (in other words, a metadata file), for example, `metadata.properties`, and
- multi-format object subdirectories.

## The `publication.properties` file

The `publication.properties` file is the object 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 object versions,
  - `pr` – access only to published object versions,
  - `pe` – object management.
- The `publication.collections` key  
Comma-separated identifiers of collections to which the given object is to be assigned.
- The `publication.destination.directoryId` key  
The identifier of the directory in which the given object is to be put.
- The `publication.metadataFile` key  
The name of the metadata file (with the extension). A metadata file must be in the directory which describes the given object 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 object. 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 object 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 `image.content` key  
The name of the file which contains the thumbnail of the given edition.
- The `publication.mainFile.FORMAT_NAME` key  
The name of the main file of the object (with the extension and the directory name which corresponds to the format of the added object). The list of supported formats is available [here](#).
- The `main.Format` key  
The name of the format indicated as the main format of the object. It must be one of the added `formats.publication.properties`.

Sample `publication.properties` file:

```
publication.properties
```

```
publication.collections=117
publication.destination.directoryId=16
publication.metadataFile=metadata.properties
publication.name=Sample multi-format file
publication.published=true

publication.mainFormat=bitmapa
publication.mainFile.PDF=PDF/pa1940-0000-00-0001.pdf
publication.mainFile.DJVU=DJVU/index.djvu
publication.mainFile.TXT=TXT/pa1940.txt
publication.mainFile.bitmapa=bitmapa/PresentationData.xml
```

In the example, the object will be assigned to the collection with identifier 117. The object will be put in the directory with identifier 16. The main format of the publication will be a **bitmap**, and the main file will be **PresentationData.xml**. The metadata will be described in the **metadata.properties** file. The object will be named **Sample multi-format file**. The object will be published. Apart from the main format (bitmap), the following formats will be added: **PDF, DjVu, and TXT**.