

[EN] 01. The Tool for Adding Files to a Planned Publication

General information

The described tool is for adding files to a planned object. The files to be added, the user in the name of whom they are to be added, and the object to which they are to be added are indicated in the configuration parameters of that tool. The tool is run from the command line and is for use in the mechanisms which automate the work of digital library editors. For example, it can be used when the following process of editorial work is defined:

1. The editor creates a planned object, by defining its properties: metadata, user permissions, and assignment to a collection.
2. The editor scans the analog form of the object to high-quality files (for example, high-resolution graphics files).
3. The editor sends the high-quality files to the automatic object entering system.
4. That system converts the files to files in a format in which the digital library presents digitized resources (for example, the PDF or DjVu format) and uses the tool for automatically adding a planned object (described in this section).

As a result, there appears a normal object – created by automatically adding files to it – in the library. Thanks to that mechanism, the editor can focus on selecting the object to be scanned and on scanning appropriate documents. The rest (from the moment the files have been scanned and sent) is done by the automation system for editors' tasks.

Every distribution version of the dLibra system has that tool. It is located in the “binaries/dlibra-client-publication- files-uploader” directory. The directory contains the following elements:

- `lib` – a directory which contains the libraries necessary for running the tool,
- `config.xml` – a file which contains information about the server to which the tool for adding files is to be connected,
- `users.xml` – information about the users in the name of whom the tool will be adding files to the planned object,
- `run.bat` – a script which runs the tool in the environment of systems from the Windows family, and
- `run.sh` – a script which runs the tool in the environment of systems from the Linux family.

In order to run the tool for adding files to a planned publication, a Java virtual machine (version 1.8 or newer) of the Sun company must be installed. Usually, it is not necessary to download the files of the tool again after the dLibra system has been updated (unless the tool stops working).

Configuring the Tool

The `config.xml` File

The `config.xml` file is an XML file which contains information about the dLibra server to which the tool is to be connected. Two properties can be defined in that file:

- the address of the dLibra server (`server` key), and
- the port on which the dLibra server listens for connections (`port` key).

The default file looks as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties>
<comment>
Configuration file for files uploader.
Properties in this file specify: server host and server port.
1. Server host
In order to specify server host place an entry which has 'server' as a key. Value specified for this key
determines the server host used by the files uploader.
2. Server port
In order to specify server port place an entry which has 'port' as a key. Value specified for this key
determines the server port used by the files uploader.
</comment>
<entry key="server">localhost</entry>
<entry key="port">10051</entry>
</properties>
```

The example shows that the tool will connect to the localhost local computer on the 10051 port.

The `users.xml` file

The `users.xml` file contains information about the users in the name of whom the tool will be adding files to the object. It is a simple XML file which defines properties, where the key is the user's login and the value is the user's account password. A sample `users.xml` file is presented below:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties>
<comment>
Configuration file for files uploader.
Properties in this file specify information about users on behalf of which this files uploader works
Each entry in this file is composed of key and value. Key indicates user login and value indicates password for
this user.
</comment>
<entry key="jkowalski">mojehaslo</entry>
</properties>
```

that file, the authentication details for one user are defined: login `jkowalski` and password `mojehaslo`.

Running the Tool

Scripts `run.bat` and `run.sh` have two parameters. The first one is the path to the main file of the object. The second one is value `true` or `false`, which determines if the newly created edition is (`true`) or is not (`false`) to be published after the files have been added.

Apart from indicating the main file, the first parameter has an additional meaning: particular elements of the path define the user in the name of whom the files are to be added and the identifier of the planned object. The path consists of the following elements:

`<PREFIX>/<USER_ID>/out/<PUB_ID>/<MAIN_FILE_NAME>`

thus:

- `<PREFIX>` is the first part of the path, immaterial from the point of view of the tool,
- `<USER_ID>` is the directory the name of which is the login of the user in the name of whom the tool is to add object files,
- `out` is the directory which contains the given user's objects,
- `<PUB_ID>` is the directory the name of which is the identifier of the planned object to which object files are to be added; that directory contains all files of the object, and
- `<MAIN_FILE_NAME>` is the name of the main file of the publication

Here is a sample execution of the script:

```
run C:\pliki\jkowalski\out\223\directory.djvu true
```

In the first parameter, the prefix is `C:\pliki`, the user login is `jkowalski`, the planned object identifier is `223`, and the main file of the object is `directory.djvu`. After the command has been executed, the tool will add all files from the `C:\pliki\jkowalski\out\223` directory to the planned object with identifier `223`. The main file of the object will be `directory.djvu`, and the object will be added in the name of user `jkowalski`.