

# [EN] 07. The Update System

## System aktualizacji

- [Updating the dLibra System](#)
- [Installing the Updater](#)
- [Before Updating](#)
- [Updating the dLibra Server](#)
- [Updating the Reader Application of the dLibra System](#)
- [Updating the Editor and Administrator Application](#)
- [Frequently Asked Questions](#)
  - [What should be done when the dLibra server runs on two machines?](#)
- `updater.properties`: optional parameters



### Skrócenie procesu aktualizacji

From version 1.1.0, the updater makes it possible to update any version to the newest one in just one step!

## Updating the dLibra System

In general, updating the dLibra system consists in copying newer file versions to the server and to the Reader Application. In the case of greater changes, it may be necessary to also edit some configuration files. Of course, the user must know which files have been changed and where they should be copied to. In order to avoid potential errors related to erroneous copying of files, we have decided to automate that process as much as possible. For that purpose, we have created the updater of the dLibra system. It is a small application which makes it possible to accelerate the update process. In this document, we discuss the general issues related to the installation and operation of the updater, and we explain how to do updates with the use of that tool.

**Please do not start up the updater before you have read the following documentation.**

## Installing the Updater

In order to install the updater of the dLibra system, you have to download the archive, in the selected format: `dlibra-jnlp-updater.zip` or `dlibra-jnlp-updater.tar.gz`. The content of the archive should be placed in any location on the computer on which the dLibra system is installed.

The updater is a Java application which makes use of the JNLP protocol. After the first start-up, all the required libraries will be downloaded to the drive, and for the following start-ups, only the possible changes (corrections) in the updater will be downloaded (not the whole application).

After the unpacking (and before the first start-up), the updater directory should contain the following files:

**`jnlp-updater.bat`**

**`jnlp-updater.sh`**

The scripts which run the updater for Windows and Linux systems. In the case of Linux system, the **`chmod +x jnlp-updater.sh`** command should be called.

**`netxjar`**

**`NETX-LICENSE.TXT`**

A minimalist implementation of the JNLP protocol which make it possible to use that protocol regardless of the Java Web Start framework of the Sun Microsystems Inc. company.

**`updater.properties`**

The updater configuration (see chapter "Before Updating").

## Before Updating

For information about new releases of the dLibra system, see [here](#).

When a new version is published in the update repository (see "The Repository of the dLibra Software", the RSS channel is also updated. After the new version of the dLibra system has been published, the update process can begin. First, the person doing the update should enter parameter values in the **`updater.properties`** file and create an account in the repository of the dLibra software. Please remember that all paths to files/directories should be entered with the use of `" / "`, regardless of the operating system.

The **`updater.properties`** file contains the following configuration parameters:

**general.serverHomeDirectory** – a directory in which the dLibra server is installed;

**general.webappHomeDirectory** – a directory in which the Reader Application of the dLibra system is installed;

**general.target**

The updater can serve two update-related purposes: **server** or **webapp**; the server mode starts up a server update (see “[Updating the dLibra Server](#)”) while the webapp mode means updating the Reader Application (see “[Updating the Reader Application](#)”);

**webdav.url** – an [address for the repository of the dLibra system](#); and

**webdav.username** and **webdav.password** – the user name and the password which make it possible for the user who is doing an update to access the repository of the dLibra system.

Before the actual update process, a backup of the currently installed version of the system should be done.

It is recommended that the dLibra server be updated first, before the Reader Application.

The system makes it possible to do an update within the framework of the same, ‘small’ version and between ‘small’ versions. The small version consists of the two first numbers of a version number, for example, for 6.1.0, the ‘small’ version is 6.1. An update within the framework of a ‘small’ version is done automatically. In order to update the system to another ‘small’ version, for example, from 6.1 to 6.2, a parameter with the required version must be passed on to the updater. If new versions of the same and of another ‘small’ version are available, the user must also indicate which version is to be installed.

An update of a particular version of is enforced with the use of parameter **updater.requestedVersion**. For example, if we have version 6.1.0 and want to update the system to the 6.2.0 version, we enter the following code in the “*updater.properties*” file:

```
updater.requestedVersion=6.2.0
```

Please remember to delete that entry from the file after the update.

The update process makes it possible to update any version to any other version, as long as there is such a possibility. In the case of an automatic update within the framework of one ‘small’ version, whenever the updater is called, an update to the latest version takes place. For example, when version 5.0.11 is installed, the updater has to be called once for the dLibra server to be updated to the newest version, for example, 6.0.24.

## Updating the dLibra Server

The server is updated in the following steps:

1. Do a backup of the currently installed version of the server.
2. End the dLibra server (and the Tomcat server).
3. In the **updater.properties** file set **general.serverHomeDirectory** and **general.target** to value **server**.
4. Call the script which starts up the updater – it should be the appropriate script for the installed operating system.

When the startup script has been called, the updating process will begin. If it is the first start-up, the updater will first download ‘itself’, and then it will begin the actual updating process. The updating process looks as follows:

1. The updater determines the parameters of the server (that information is taken mainly from databases; the updater uses information from the **data base.properties** file of the server):
  - a. Which services are installed?
  - b. Does the server work?
  - c. Which version of the dLibra system is currently installed?
2. Are there newer versions in the repository?
  - a. If not, the updater is ended.
3. The updater connects with the repository of the dLibra software, which contains updates, and downloads the files necessary for the update.
4. The status of the dLibra server is checked:
  - a. If the dLibra server is running, the updater will wait until it is ended.
  - b. If it is ended, the updater will move to the next step of the update.
5. Old files are removed and replaced with new ones.
6. If the person who is doing the update has to intervene, the information about it will be displayed on the console.
  - a. For each such file:
    - i. copy of the old file with the “fromVersion\_x\_x\_x\_” extension (where x\_x\_x\_ is an old version of the dLibra system, for example, pages.xml.fromVersion\_5\_0\_1\_) will be created;
    - ii. the old file will be replaced with a new one.
7. The update files downloaded to the temporary directory are deleted.
8. At the end of the server update, the program displays information about which server services and Reader Applications are still to be updated (if the dLibra server is installed on more than one computer, the updating process has to be repeated for each of them).

9. That ends the process of the automatic update of the server; if there are newer versions in the repository, the updater will display an appropriate message.

After the updater has finished its job, the user should review the files indicated by the program as requiring an intervention of the person who is doing the update. After the server has been updated to the latest version, the user can proceed to update the Reader Application.

## Updating the Reader Application of the dLibra System

The update process of the Reader Application is similar to the update process of the dLibra server, described in the previous section.

1. Do a backup of the currently installed version of the Reader Application.
2. End the Tomcat server (if it has not been ended yet).
3. In the **updater.properties** file set **general.webappHomeDirectory** and **general.target** to value **webapp**.
4. Call the script which starts up the updater – it should be the appropriate script for the installed operating system.

The steps of the update process for the Reader Application are very similar to those for the server (some steps are omitted). After the server and Reader Application have been updated, the dLibra system can be started up.

## Updating the Editor and Administrator Application

When the Reader Application has been updated, all installations of the Editor and Administrator Application will be updated during the next start-up.

## Frequently Asked Questions

### What should be done when the dLibra server runs on two machines?

If the dLibra server runs on several servers, the updater should be installed on each of them, and the update process should be repeated for every instance of the server. The same principle applies to a situation in which the Reader Application is not installed on the dLibra server but on a different machine.

### updater.properties: optional parameters

**server.checkRunning** – if that parameter has value **false** (the default value is **true**), the updater will not check if the server is running. The **false** value may be helpful if we want to update copies of files of the Reader Application and of the server. In such a case, there is no need to switch off any of the elements of the dLibra system, and after the update process has finished, old files can be replaced with new ones.

**delete.downloaded.patch** – by default, the updater assumes the **true** value of that parameter. The updater downloads the update files from the repository of the dLibra software. With the default values, the updater deletes all the downloaded files just before it finishes its job. If we want to retain the downloaded update, value **false** should be set. Regardless of the value set for that parameter, that parameter will be cleared during the next start-up of the updater to which the updates are downloaded.

**general.tempDirectoryPath** – by default, the updater downloads files to the **dLibraUpdater** directory, which is created in a location indicated by the **java.io.tmp** system variable (it is a variable of a Java virtual machine). In the case of Linux systems, that directory usually corresponds to the value of the **\$TEMP** variable. In order to modify the directory in which the updates are saved, the **general.tempDirectoryPath** key should be added to the **updater.properties** file, and the desired directory should be given as the value.