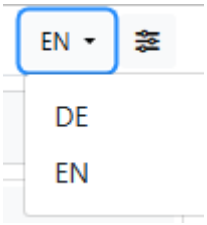
The DLNA/UpNP Media Browser

The DLNA Media Browser is a file explorer for files offered via a dlna/upnp service. In the current version of the tool it is possible to view the folder structure and files on the server and download selected files one by one. You cannot edit or delete the files on the server.

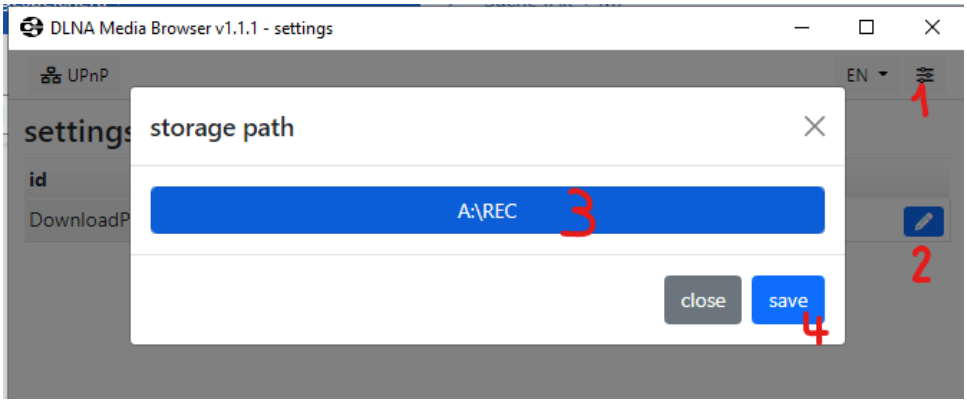
Change language

In the current version, the language English or German can be set. The language can be changed in the header navigation and is remembered even after the application is closed.



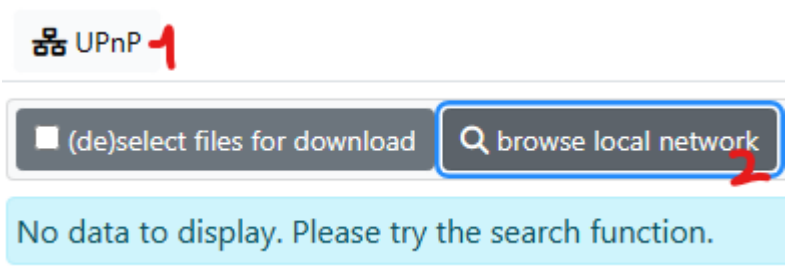
Configuration of the storage path for downloaded files.

Unless otherwise configured, downloaded files are stored in the default download directory: "C:\Users\username\Downloads". With the settings icon (1) you open the configuration view, in which the storage path can be changed. The memory path is remembered even after the application is closed.

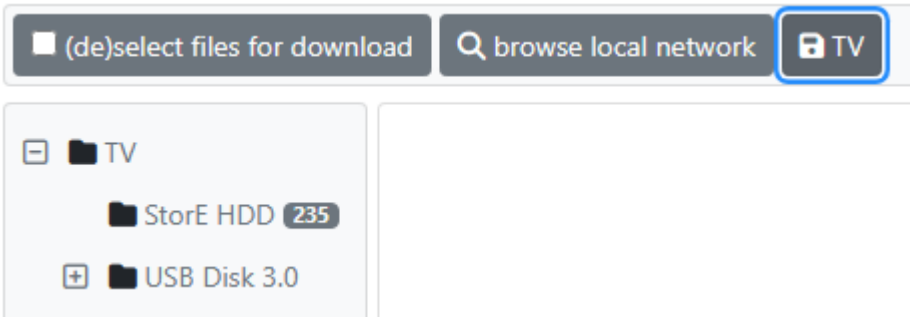


Find and browse dlna/upnp devices on the local network.

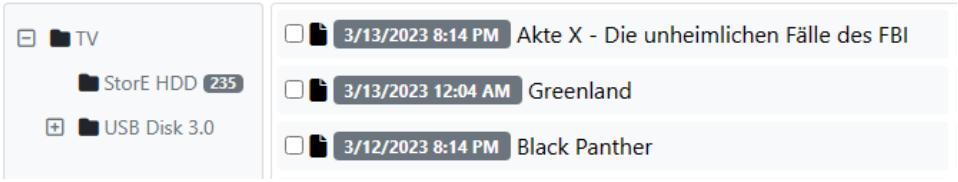
Clicking on the UPnP icon (1) opens the browser view. The button with the magnifying glass symbol (2) searches the local network and lists all found dlna/upnp devices as a new button to the right of the search button.



In the example, a TV set could be found. Clicking on the button with the device name starts a search for all data carriers connected to the device and the files contained therein. For example, a recording hard disk and a USB stick are connected to the TV, which are displayed as separate subfolders of the TV.

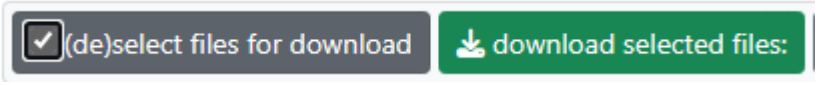


A label next to the folder name shows the number of files contained in the folder. A click on the folder name opens a list with the files it contains.

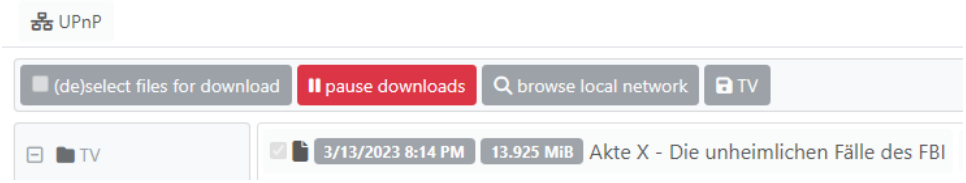


Download files.

The download button is only visible when at least one file is selected. Individual files can be selected by clicking on the file name. Via the button in the menu bar all files in the current folder can be selected at once.



The download will download the selected files one by one. Leaving the view leads to the abortion of the current download, therefore all buttons except the pause button are deactivated during the download.

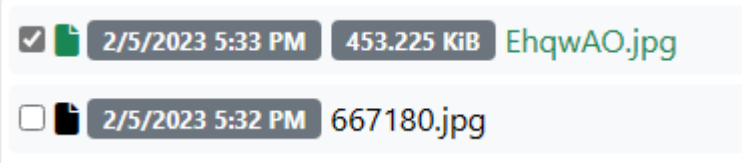


The label next to the file name indicates the download progress in terms of the already downloaded file size. Unfortunately, the information for the total file size of streams is not always available, so no progress of 0-100% can be displayed at this point. Maybe a solution can be found in the upcoming versions of the application.

The download can be cancelled at any time by clicking on the pause button. Files that could not be fully downloaded are highlighted in red.

If a download is canceled, the rest of the download queue is also suspended. Incompletely downloaded files are automatically removed from the hard disk and cannot continue. This is to ensure that no corrupt or incomplete downloads end up on the hard drive unnoticed.

During the download, the file has the extension .part and is renamed after a successful download with the correct file extension. If a download is successful, the file name is highlighted in green.

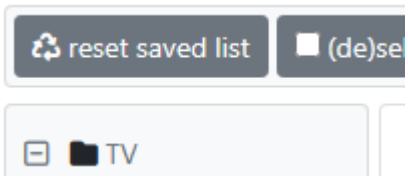


Fully downloaded files will not be downloaded again as long as they remain in the download directory. The downloader detects complete files and skips them if they are selected again for download.

Each file is stored in a subfolder that has the same name as the file. The name of the file is supplemented by a date text. When downloading TV recordings of a series, it has the advantage that all recordings of the same series can be found in a common folder sorted by recording date.

File List Caching

Searching the upnp/dlna devices for existing files can take some time, so I decided to cache the last list I found. This list is immediately available again after closing and opening the application. Men detects that a previously saved state of the list is open when the reset button is visible. By clicking on this button, the saved list can be reset.



Updates / New Versions

When there are new updates, an update button appears next to the language dropdown. By clicking on this button, the new version of (https://github.com/hswlab/dist-dlna-browser-.net/releases) will be downloaded. The download button should show a download progress with a little delay. Once the download is complete, the application must be closed. The update will then be installed automatically. The download feedback is unfortunately a bit sluggish. If you want to see a more accurate progress bar, you could also download the new version manually from GitHub via your browser.

My versioning consists of 3 digits e.g.: "1.2.3" The last digit announces a bugfix, the middle one a new feature and the first digit a more extensive change.

What is this app programmed with?

This is an Electron application in combination with a . NET6 MVC framework (electron.net). The business logic is mainly programmed with C#. In the frontend, Bootstrap and classic CSS are used. A bit of JQuery and Ajax were also used for frontend interactions. The local LiteDB database is used to persist user settings and the file list. The data is only stored locally, communication with the Internet

does not take place! The library LibVLCSharp is used to determine the UPnP services in the local network and read the files.

Will there be a version for Linux?

Since it is an Electron application, a version for Linux or Mac OS can theoretically be provided. However, I have not yet tested whether the LibVLCSharp library works under Linux. But I will definitely deal with it. For now, however, it is important to me that the Windows version works without any problems. So far, this app has only been tested with a Panasonic TV.

Known Issues

When clicking the device button, it can sometimes happen that not all files from the disks of the DLNA/UPnP device are listed. I'm still not sure if the problem still exists, or if I've already fixed it. At least I couldn't recreate this bug. Try to click on the device button several times, if it should happen, the files would have to appear at some point. I think the problem is that LibVLCSharp communicates asynchronously with the server and my application and there doesn't seem to be any reliable feedback telling me when all the data on the server has been found completely.

What features are planned next?

A filter option would be useful to find certain files faster. Perhaps a grouped display of files of the same name in the browser would be interesting (useful for displaying TV recordings of a series). In this regard, a filter on similarly written files would also be useful.

With the LibVLCSharp library you would theoretically also have the possibility to play audio or video files directly in the application without having to download them first. I would also like to try some Electron features, such as the fullscreen function or minimizing to a taskbar icon.

Downloads are currently stored in subfolders grouped by name. You could also offer alternative storage behavior in the settings.

Unfortunately, the Panasonic TV does not seem to give the LibVLCSharp library any information about which file extension a recording has and how big it is. This makes it difficult to determine the download progress and save downloads with the correct file extension. Streams are currently stored as .ts (Transport Stream). I hope to find a more reliable solution to this issue.