MPL-Data-Cast

Release 0.6.1.post1+g163c1cd

Paul Müller and others

Jan 06, 2024

CONTENTS:

CHAPTER

ONE

INTRODUCTION

1.1 Installation

For Windows and macOS users, there are installers available at the release page. The Windows installer also includes the *command line interface (CLI)* `mpldc.exe`. You might want to add the installation directory to your Windows PATH variable for conveniently using the CLI (see e.g. this guide).

If you have Python installed, you can install MPL-Data-Cast with:

```
pip install mpl_data_cast
```

The *Command-line interface* is available via:

```
mpldc
# or
python -m mpl_data_cast.cli
```

To start the GUI, run:

```
mpldc-gui
# or
python -m mpl_data_cast.gui
```

1.2 Issue tracker

If you encounter bugs or have suggestions for improvement, open an issue on GitHub.

1.3 Motivation

If you do a lot of experiments, you have a lot of data. If you want to analyze the data on a different computer, you have to make sure that the data are transferred correctly to that computer. MPL-Data-Cast can help you with that process and it also allows you to simplify the subsequent data analysis by reformatting your raw input data. The main features of MPL-Data-Cast are:

- Copy a directory tree from a local file system to a network share (or any other mountable location) and verify that all files are copied correctly via comparison of MD5 checksums
- Convert your raw data to a file format that your data analysis pipeline understands (e.g. convert TIF files to HDF5 files) or repack/compress your input data

- Append additional meta data from your experiments that you were not able to set in your data acquisition software

MPL-Data-Cast is developed at the Max Planck Institute for the Science of Light, where we have a lot of such data. For us, it addresses several data management issues:

- When you copy a directory tree containing large (~20GB) files from a local Windows machine to a remote network share using the built-in File Explorer, sometimes the files were corrupted (e.g. HDF5 files could not be opened anymore).
- When you deal with RT-DC data, you always have to (losslessly) compress the data (or at least repack the file so that a dataset is not scattered over the entire HDF5 file) to properly work with it.
- We also have very basic data acquisition pipelines that work with Micro-Manager which outputs TIF files and metadata files. We needed to merge these files into properly formatted (including metadata) HDF5 files.

1.4 Design

MPL-Data-Cast is a Python library with a command-line interface (CLI) and a very simple GUI on top that lets you apply a “recipe” to data files. A recipe defines how your acquisition data is transformed into the final raw data for your analysis pipeline. You can define your own recipes or use the recipes that come with MPL-Data-Cast.

See also:

- *Command-line interface*
- *Graphical user interface*

1.5 Good to know

- MPL-Data-Cast does not create a directory with the same name as the input directory in the output directory, it copies the content of the input directory into the output directory. That means that you have to be careful when entering the output path, or otherwise you might end up mixing up data.
- Right now, it is not possible to copy only selected files or subfolders from a given directory, you can only transfer complete directories.

CHAPTER

TWO

COMMAND-LINE INTERFACE (CLI)

The `mpldc` CLI gives you full access to the MPL-Data-Cast core functionalities. This section documents the output of the different commands (e.g. `mpldc --help` or `mpldc cast --help`).

2.1 mpldc

```
mpldc [OPTIONS] COMMAND [ARGS]...
```

Commands

cast
Convert and copy experimental data

list-recipes
List available recipes that can be used in the `mpldc cast` subcommand

2.2 mpldc cast

Cast data from a source directory to a target directory

This will convert all data under the tree in PATH_RAW and copy them to PATH_TARGET.

```
mpldc cast [OPTIONS] PATH_RAW PATH_TARGET
```

Options

-r, --recipe <recipe>
specifies recipe to use, defaults to 'CatchAll'; see the `mpldc list-recipes` subcommand for more options

-o, --options <options>
comma-separated keyword arguments passed to the recipe's *convert_dataset* method, e.g. wavelength=984e-9,pixel_size=1.2e-6

Arguments

PATH_RAW
Required argument

PATH_TARGET
Required argument

2.3 mpldc list-recipes

List available recipes that can be used in the `mpldc cast` subcommand

```
mpldc list-recipes [OPTIONS]
```

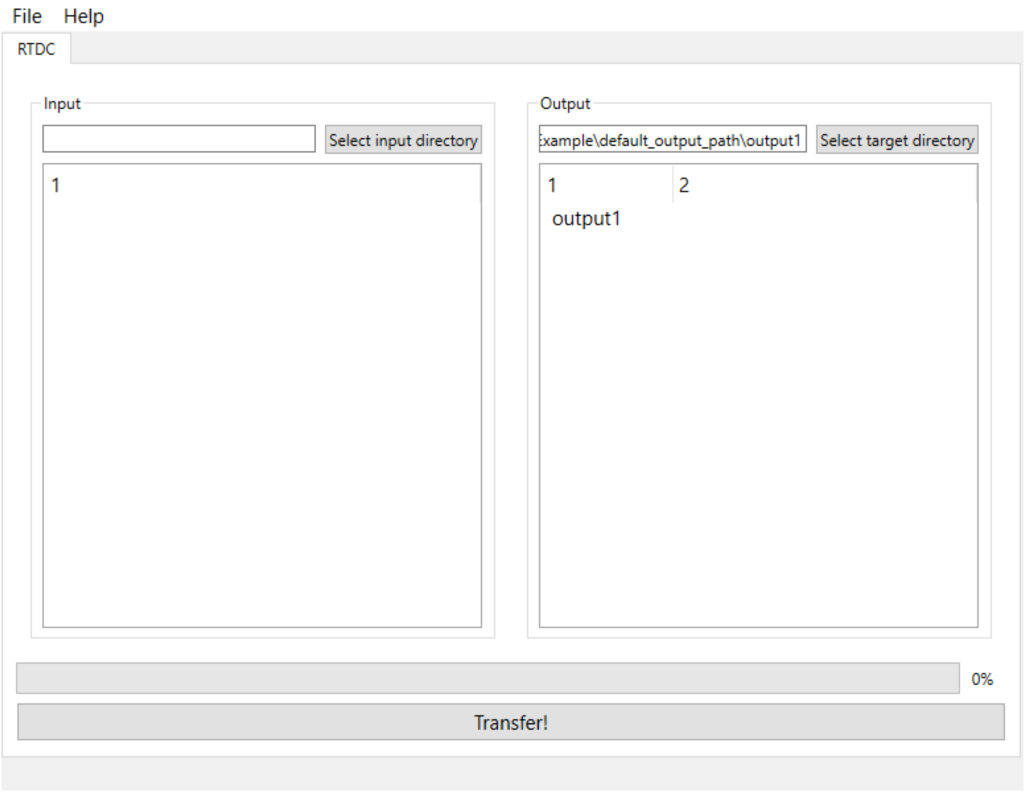
CHAPTER

THREE

GRAPHICAL USER INTERFACE (GUI)

The GUI is an easy to use way to use MPL-Data-Cast. The main window consists of a tab for data of each experiment type, like, e. g. RT-DC.

3.1 Main window



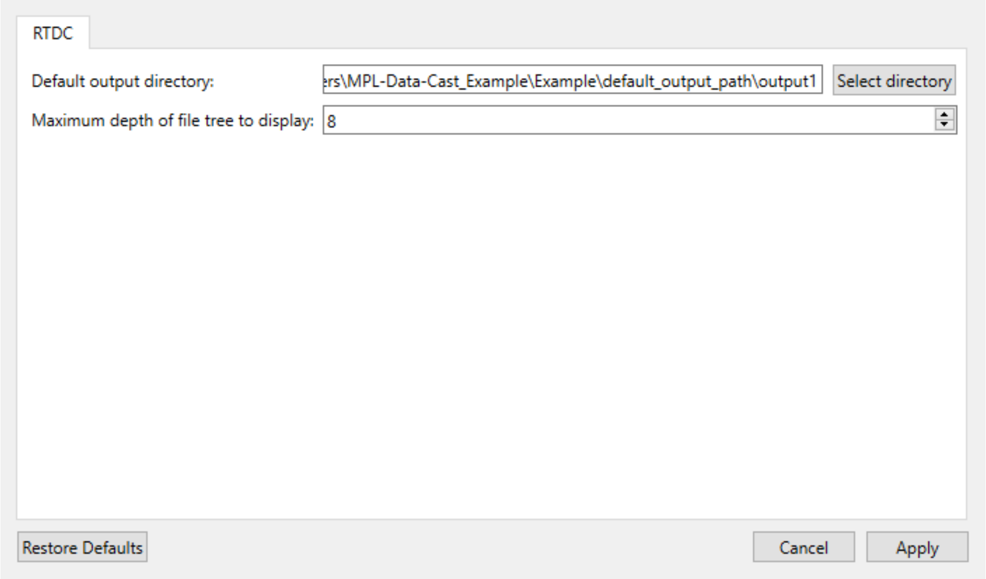
On the left, you can specifiy the input directory, either by dragging-and-dropping a folder or by using the button "Select input directory". You can also drag-and-drop a file, in that case, the parent directory of that file will be used. On the right, the output directory is shown, it can be changed in the same way as the input directory.

For convenience, the default output directory can be changed in the settings.

The two big white areas will show the contents of input and output directory respectively. Since large directories with many subdirectories can cause performance issues, one can change the maximum depth of directories that should be displayed here. The minimum is 1 and the maximum is 24.

The button at the bottom of the window starts the transfer of data using the RT-DC recipe. The progress bar indicates the current status of the file transfer and is updated after each file that was processed.

3.2 Preferences



Default output directory: rs\MPL-Data-Cast_Example\Example\default_output_path\output1

Maximum depth of file tree to display: 8

In the top, there is current default output directory shown. It can be changed via the button "Select directory".

Below that, the maximum depth of subdirectories that will be shown in the main window can be set. The minimum is 1, so that at least the direct contents of input and output directories are shown. The maximum is 24.

Warning: Using a large depth limit for displaying sub-directories can cause performance issues, especially if the directories contain a lot of files.

CHAPTER

FOUR

INDICES AND TABLES

- genindex
- modindex
- search

INDEX