
mSP2lib

Release 0.1.3

Convert an MSP file representing one or more Mass Spectra to a NIST MS Search user library.

Dominic Davis-Foster

Apr 08, 2026

Contents

1	Installation	3
1.1	from PyPI	3
1.2	from GitHub	3
1.3	API Reference	3
1.4	msp2lib	5
1.5	Contributing	6
1.6	License	7
1.7	Downloading source code	11
	Python Module Index	13
	Index	15

The NIST Library Conversion program running in Wine and Docker

This is a Python script to facilitate conversion from MSP files to NIST MS Search user libraries.

Installation

1.1 from PyPI

```
$ python3 -m pip install msp2lib --user
```

1.2 from GitHub

```
$ python3 -m pip install git+https://github.com/domdfcoding/msp2lib@master --user
```

1.3 API Reference

Table of Contents

- *API Reference*
 - `msp2lib.core`
 - `msp2lib.utils`

1.3.1 `msp2lib.core`

Convert an MSP file representing one or more Mass Spectra to a NIST MS Search user library.

Docker must be installed to use this program.

The first time this script is run it will download the latest version of the docker image automatically.

This can also be done manually, such as to upgrade to the latest version, by running with the `--get-docker-image` flag.

Functions:

<code>main()</code>	Entry point for running from the command line.
<code>msp2lib(msp_file, output_dir[, lib_name])</code>	Convert the provided MSP file to a NIST User Library, and store the newly created library in the given output directory.

`main()`

Entry point for running from the command line.

misp2lib(*misp_file*, *output_dir*, *lib_name=None*)

Convert the provided MSP file to a NIST User Library, and store the newly created library in the given output directory.

Parameters

- **misp_file** (`Union[str, Path, PathLike]`) – The MSP file to convert to a NIST User Library
- **output_dir** (`Union[str, Path, PathLike]`) – The directory to store the NIST User Library in
- **lib_name** (`Optional[str]`) – The name of the NIST User Library. If `None` this will be the filename of the MSP file without the extension. Default `None`.

1.3.2 misp2lib.utils

utilities for misp2lib.

Functions:

<code>about()</code>	Prints information about misp2lib.
<code>build_docker_image()</code>	Build the lib2nist-wine docker image from the Dockerfile.
<code>download_docker_image()</code>	Pull the lib2nist-wine docker image from dockerhub.
<code>subprocess_with_log(command)</code>	The <code>command</code> with <code>subprocess.Popen</code> , printing any stdout from the command.
<code>test_docker()</code>	Returns whether Docker is installed on the system.
<code>version()</code>	Prints the version number of misp2lib.

`about()`

Prints information about misp2lib.

Return type `str`

`build_docker_image()`

Build the lib2nist-wine docker image from the Dockerfile.

Return type `int`

Returns The return code of the `docker build` command

`download_docker_image()`

Pull the lib2nist-wine docker image from dockerhub.

Return type `int`

Returns The return code of the `docker pull` command

`subprocess_with_log(command)`

The `command` with `subprocess.Popen`, printing any stdout from the command.

Parameters `command` (`Union[str, Sequence[str]]`) – The command to run

Return type `Popen`

Returns

test_docker()

Returns whether Docker is installed on the system.

Return type `bool`

Returns `True` if Docker is installed, `False` otherwise.

version()

Prints the version number of msp2lib.

Return type `str`

1.4 msp2lib

Convert an MSP file representing one or more Mass Spectra to a NIST MS Search user library.

Docker must be installed to use this program.

1.4.1 Synopsis

```
msp2lib [-h] [input_file] [output_dir]
```

1.4.2 Description

Convert an MSP file representing one or more Mass Spectra to a NIST MS Search user library.

positional arguments:

input_file The path of the MSP file to be converted into a NIST User Library.

output_dir The path to the directory to save the output library in. If unspecified the current working directory is used instead.

optional arguments:

-h, --help Show usage information and exit.

--version Show the version number and exit.

--get-docker-image Download the docker image now rather than at first run, then exit. This can also be used to update the docker image.

--build-docker-image Build the docker image from the Dockerfile, then exit. This can be useful when installing offline.

1.4.3 Examples

msp2lib spectra/example.msp output/ Convert the file `example.msp` in the directory `spectra` to a library, which is saved in the directory `output`.

1.4.4 Installation

Before installing `misp2lib`, ensure you have a working Docker installation. For more information visit <https://docs.docker.com/get-docker/> for more information.

`misp2lib` can then be installed with `pip`:

```
$ pip install misp2lib
```

Once installed, `misp2lib` can be run by typing:

```
$ misp2lib
```

If `misp2lib` is not installed in a directory in `$PATH`, you may need to add `~/local/bin/` to your `$PATH`.

1.4.5 Reporting Bugs

Please use the GitHub issue tracker to report bugs: <<https://github.com/domdfcoding/misp2lib/issues>>

1.5 Contributing

`misp2lib` uses `tox` to automate testing and packaging, and `pre-commit` to maintain code quality.

Install `pre-commit` with `pip` and install the git hook:

```
$ python -m pip install pre-commit
$ pre-commit install
```

1.5.1 Coding style

`formate` is used for code formatting.

It can be run manually via `pre-commit`:

```
$ pre-commit run formate -a
```

Or, to run the complete autoformatting suite:

```
$ pre-commit run -a
```

1.5.2 Automated tests

Tests are run with `tox` and `pytest`. To run tests for a specific Python version, such as Python 3.10:

```
$ tox -e py310
```

To run tests for all Python versions, simply run:

```
$ tox
```

1.5.3 Type Annotations

Type annotations are checked using `mypy`. Run `mypy` using `tox`:

```
$ tox -e mypy
```

1.5.4 Build documentation locally

The documentation is powered by Sphinx. A local copy of the documentation can be built with `tox`:

```
$ tox -e docs
```

1.6 License

msp2lib is licensed under the [GNU Lesser General Public License v3.0](#)

Permissions of this copyleft license are conditioned on making available complete source code of licensed works and modifications under the same license or the GNU GPLv3. Copyright and license notices must be preserved. Contributors provide an express grant of patent rights. However, a larger work using the licensed work through interfaces provided by the licensed work may be distributed under different terms and without source code for the larger work.

Permissions

- Commercial use – The licensed material and derivatives may be used for commercial purposes.
- Modification – The licensed material may be modified.
- Distribution – The licensed material may be distributed.
- Patent use – This license provides an express grant of patent rights from contributors.
- Private use – The licensed material may be used and modified in private.

Conditions

- License and copyright notice – A copy of the license and copyright notice must be included with the licensed material.
- Disclose source – Source code must be made available when the licensed material is distributed.
- State changes – Changes made to the licensed material must be documented.
- Same license (library) – Modifications must be released under the same license when distributing the licensed material. In some cases a similar or related license may be used, or this condition may not apply to works that use the licensed material as a library.

Limitations

- Liability – This license includes a limitation of liability.
- Warranty – This license explicitly states that it does NOT provide any warranty.

[See more information on choosealicense.com](https://choosealicense.com) ⇒

GNU LESSER GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <<https://fsf.org/>>
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates
the terms and conditions of version 3 of the GNU General Public
License, supplemented by the additional permissions listed below.

0. Additional Definitions.

As used herein, "this License" refers to version 3 of the GNU Lesser
General Public License, and the "GNU GPL" refers to version 3 of the GNU
General Public License.

"The Library" refers to a covered work governed by this License,
other than an Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided
by the Library, but which is not otherwise based on the Library.
Defining a subclass of a class defined by the Library is deemed a mode
of using an interface provided by the Library.

A "Combined Work" is a work produced by combining or linking an
Application with the Library. The particular version of the Library
with which the Combined Work was made is also called the "Linked
Version".

The "Minimal Corresponding Source" for a Combined Work means the
Corresponding Source for the Combined Work, excluding any source code
for portions of the Combined Work that, considered in isolation, are
based on the Application, and not on the Linked Version.

The "Corresponding Application Code" for a Combined Work means the
object code and/or source code for the Application, including any data
and utility programs needed for reproducing the Combined Work from the
Application, but excluding the System Libraries of the Combined Work.

1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License
without being bound by section 3 of the GNU GPL.

2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a
facility refers to a function or data to be supplied by an Application
that uses the facility (other than as an argument passed when the
facility is invoked), then you may convey a copy of the modified
version:

(continues on next page)

(continued from previous page)

a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or

b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.

3. Object Code Incorporating Material from Library Header Files.

The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the object code with a copy of the GNU GPL and this license document.

4. Combined Works.

You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the Combined Work with a copy of the GNU GPL and this license document.

c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.

d) Do one of the following:

⓪) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the

(continues on next page)

manner specified by section 6 of the GNU GPL for conveying Corresponding Source.

1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.

e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)

5. Combined Libraries.

You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

- a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.
- b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser

(continued from previous page)

General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.

1.7 Downloading source code

The msp2lib source code is available on GitHub, and can be accessed from the following URL: <https://github.com/domdfcoding/msp2lib>

If you have git installed, you can clone the repository with the following command:

```
$ git clone https://github.com/domdfcoding/msp2lib
```

```
Cloning into 'msp2lib'...
remote: Enumerating objects: 47, done.
remote: Counting objects: 100% (47/47), done.
remote: Compressing objects: 100% (41/41), done.
remote: Total 173 (delta 16), reused 17 (delta 6), pack-reused 126
Receiving objects: 100% (173/173), 126.56 KiB | 678.00 KiB/s, done.
Resolving deltas: 100% (66/66), done.
```

Alternatively, the code can be downloaded in a 'zip' file by clicking:

Clone or download → *Download Zip*

1.7.1 Building from source

The recommended way to build msp2lib is to use tox:

```
$ tox -e build
```

The source and wheel distributions will be in the directory dist.

If you wish, you may also use pep517.build or another PEP 517-compatible build tool.

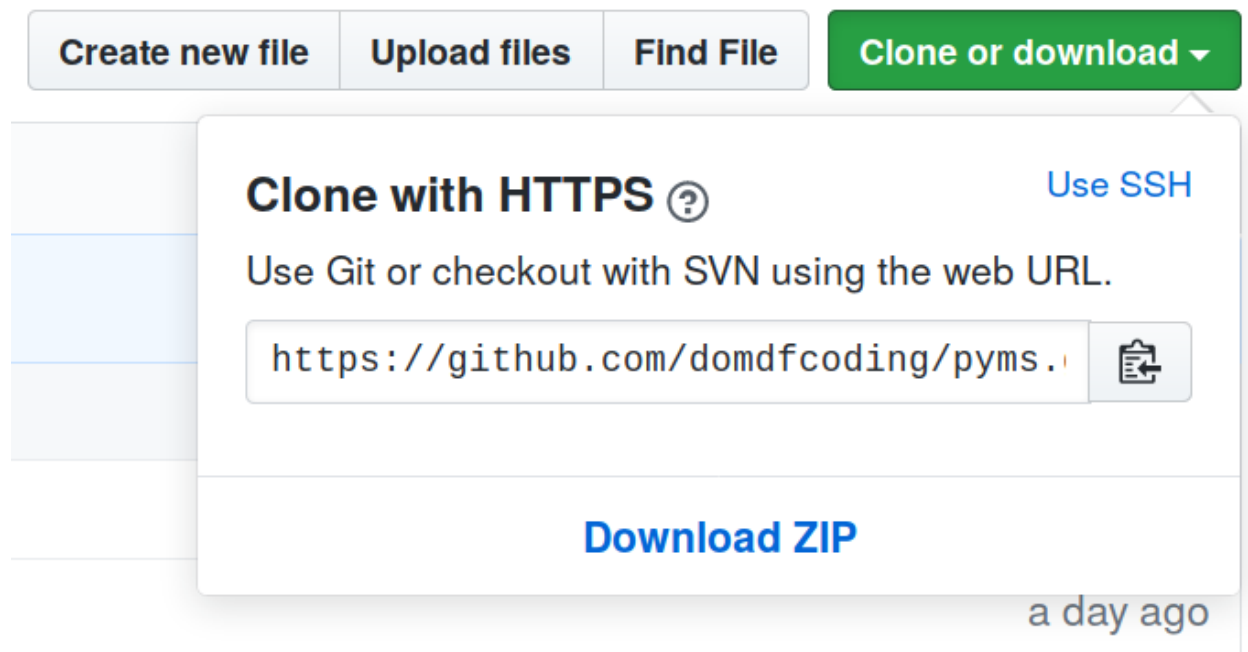


Fig. 1: Downloading a 'zip' file of the source code

Python Module Index

m

`msp2lib.core`, 3

`msp2lib.utils`, 4

A

`about()` (*in module `msp2lib.utils`*), 4

B

`build_docker_image()` (*in module `msp2lib.utils`*), 4

D

`download_docker_image()` (*in module `msp2lib.utils`*),
4

G

GNU Lesser General Public License v3.0, 7

M

`main()` (*in module `msp2lib.core`*), 3

module

`msp2lib.core`, 3

`msp2lib.utils`, 4

`msp2lib()` (*in module `msp2lib.core`*), 3

`msp2lib.core`

 module, 3

`msp2lib.utils`

 module, 4

P

Python Enhancement Proposals

 PEP 517, 11

S

`subprocess_with_log()` (*in module `msp2lib.utils`*), 4

T

`test_docker()` (*in module `msp2lib.utils`*), 5

V

`version()` (*in module `msp2lib.utils`*), 5