

---

# **msg***parserDocumentation*

***Release 1.2.0***

**Vikram Arsid**

**Mar 19, 2022**



---

## Contents:

---

<b>1</b>	<b>msg_parser</b>	<b>1</b>
1.1	Features . . . . .	1
1.2	Installation . . . . .	1
1.3	Usage . . . . .	2
<b>2</b>	<b>Installation</b>	<b>3</b>
2.1	Stable release . . . . .	3
2.2	From sources . . . . .	3
<b>3</b>	<b>Usage</b>	<b>5</b>
<b>4</b>	<b>msg_parser</b>	<b>7</b>
4.1	msg_parser package . . . . .	7
<b>5</b>	<b>Contributing</b>	<b>11</b>
5.1	Types of Contributions . . . . .	11
5.2	Get Started! . . . . .	12
5.3	Pull Request Guidelines . . . . .	13
5.4	Tips . . . . .	13
5.5	Deploying . . . . .	13
<b>6</b>	<b>Credits</b>	<b>15</b>
6.1	Development Lead . . . . .	15
6.2	Contributors . . . . .	15
<b>7</b>	<b>History</b>	<b>17</b>
7.1	0.1.0 (2018-10-26) . . . . .	17
7.2	1.0.0 (2018-10-29) . . . . .	17
7.3	1.2.0 (2019-12-12) . . . . .	17
<b>8</b>	<b>Indices and tables</b>	<b>19</b>
	<b>Python Module Index</b>	<b>21</b>
	<b>Index</b>	<b>23</b>



Python module for parsing outlook msg files.

- Free software: BSD license
- Documentation: <https://msg-parser.readthedocs.io>.

## 1.1 Features

- Parse MSG file.
- Convert MSG file to EML file.
- Output MSG file as JSON string.
- Handles nested MSG/EML attachments.
- Works 100% on Linux machines, do not require any windows libraries.

## 1.2 Installation

- Basic installation

```
pip install msg_parser
```

- With RTF decompression

```
pip install msg_parser[rtf]
```

## 1.3 Usage

- Run CLI command

```
$ msg_parser --help
usage: msg_parser [-h] -i FILE [-j] [-e EML_FILE]

Microsoft Message Parser

optional arguments:
  -h, --help            show this help message and exit
  -i FILE, --input FILE  msg file path
  -j, --json            output parsed msg as json to console
  -e EML_FILE, --eml EML_FILE
                        provide email file path to save as eml file.
```

- Import in python modules

```
from msg_parser import MsOxMessage

msg_obj = MsOxMessage(msg_file_path)

json_string = msg_obj.get_message_as_json()

msg_properties_dict = msg_obj.get_properties()

saved_path = msg_obj.save_email_file(output_eml_file_path)
```

### 2.1 Stable release

To install `msg_parser`, run this command in your terminal:

```
$ pip install msg_parser
```

This is the preferred method to install `msg_parser`, as it will always install the most recent stable release.

If you don't have `pip` installed, this [Python installation guide](#) can guide you through the process.

### 2.2 From sources

The sources for `msg_parser` can be downloaded from the [Github repo](#).

You can either clone the public repository:

```
$ git clone git://github.com/vikramarsid/msg_parser
```

Or download the [tarball](#):

```
$ curl -OL https://github.com/vikramarsid/msg_parser/tarball/master
```

Once you have a copy of the source, you can install it with:

```
$ python setup.py install
```





## CHAPTER 3

---

### Usage

---

To use msg\_parser in a project:

```
from msg_parser import MsOxMessage

msg_obj = MsOxMessage(msg_file_path)

json_string = msg_obj.get_message_as_json()

msg_properties_dict = msg_obj.get_properties()

saved_path = msg_obj.save_email_file(output_eml_file_path)
```



## 4.1 msg\_parser package

### 4.1.1 Subpackages

msg\_parser.properties package

#### Submodules

msg\_parser.properties.ms\_props\_date\_type\_map module

msg\_parser.properties.ms\_props\_generator module

msg\_parser.properties.ms\_props\_generator.**generate\_data\_id\_type\_mapping**(*master\_map*)

msg\_parser.properties.ms\_props\_generator.**generate\_id\_name\_mapping**(*master\_map*)

msg\_parser.properties.ms\_props\_generator.**generate\_master\_properties**()

msg\_parser.properties.ms\_props\_id\_map module

#### Module contents

### 4.1.2 Submodules

### 4.1.3 msg\_parser.cli module

Console script for msg\_parser.

```
class msg_parser.cli.FullPaths (option_strings, dest, nargs=None, const=None, default=None,
                                type=None, choices=None, required=False, help=None,
                                metavar=None)

    Bases: argparse.Action

    Expand user- and relative-paths

msg_parser.cli.create_parser (args)

msg_parser.cli.is_dir (dir_name)
    Checks if a path is an actual directory

msg_parser.cli.main ()
```

#### 4.1.4 msg\_parser.data\_models module

```
class msg_parser.data_models.DataModel
    Bases: object

    static PtypBinary (data_value)
    static PtypBoolean (data_value)
    static PtypCurrency (data_value)
    static PtypErrorCode (data_value)
    static PtypFloating32 (data_value)
    static PtypFloating64 (data_value)
    static PtypFloatingTime (data_value)
    static PtypGuid (data_value)
    static PtypInteger16 (data_value)
    static PtypInteger32 (data_value)
    static PtypInteger64 (data_value)
    static PtypMultipleBinary (data_value)
    static PtypMultipleCurrency (data_value)
    static PtypMultipleFloating32 (data_value)
    static PtypMultipleFloating64 (data_value)
    static PtypMultipleFloatingTime (data_value)
    static PtypMultipleGuid (data_value)
    static PtypMultipleInteger16 (data_value)
    static PtypMultipleInteger32 (data_value)
    static PtypMultipleInteger64 (data_value)
    static PtypMultipleString (data_value)
    static PtypMultipleString8 (data_value)
    static PtypMultipleTime (data_value)
    static PtypNull (_)
    static PtypObject (data_value)
```

```
static PtypRestriction (data_value)
static PtypRuleAction (data_value)
static PtypServerId (data_value)
static PtypString (data_value)
static PtypString8 (data_value)
static PtypTime (data_value)
static PtypUnspecified (data_value)
get_value (data_value, data_type_name=None, data_type=None)
static lookup_data_type_name (data_type)
msg_parser.data_models.get_floating_time (data_value)
msg_parser.data_models.get_multi_value_offsets (data_value)
msg_parser.data_models.get_time (data_value)
```

#### 4.1.5 msg\_parser.email\_builder module

```
class msg_parser.email_builder.EmailFormatter (msg_object)
    Bases: object

    build_email ()

    save_file (file_path, file_name=None)
msg_parser.email_builder.flatten_list (string_list)
msg_parser.email_builder.normalize (input_str)
```

#### 4.1.6 msg\_parser.msg\_parser module

```
class msg_parser.msg_parser.Attachment (attachment_properties)
    Bases: object

    class to store attachment attributes

class msg_parser.msg_parser.Message (directory_entries)
    Bases: object

    Class to store Message properties

    as_dict ()
        returns message attributes as a python dictionary. :return: dict

class msg_parser.msg_parser.MsOxMessage (msg_file_path)
    Bases: object

    Base class for Microsoft Message Object

    get_email_mime_content ()

    get_message_as_json ()

    get_properties ()

    get_properties_as_dict ()
```

```
is_valid_msg_file()  
save_email_file(file_path, file_name=None)  
class msg_parser.msg_parser.Recipient(recipients_properties)  
    Bases: object  
    class to store recipient attributes  
msg_parser.msg_parser.format_size(num, suffix='B')  
msg_parser.msg_parser.parse_email_headers(header, raw=False)
```

### 4.1.7 Module contents

Top-level package for msg\_parser.

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given. You can contribute in many ways:

## 5.1 Types of Contributions

### 5.1.1 Report Bugs

Report bugs at [https://github.com/vikramarsid/msg\\_parser/issues](https://github.com/vikramarsid/msg_parser/issues).

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

### 5.1.2 Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with “bug” and “help wanted” is open to whoever wants to implement it.

### 5.1.3 Implement Features

Look through the GitHub issues for features. Anything tagged with “enhancement” and “help wanted” is open to whoever wants to implement it.

### 5.1.4 Write Documentation

msg\_parser could always use more documentation, whether as part of the official msg\_parser docs, in docstrings, or even on the web in blog posts, articles, and such.

### 5.1.5 Submit Feedback

The best way to send feedback is to file an issue at [https://github.com/vikramarsid/msg\\_parser/issues](https://github.com/vikramarsid/msg_parser/issues).

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

## 5.2 Get Started!

Ready to contribute? Here's how to set up msg\_parser for local development.

1. Fork the msg\_parser repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/msg_parser.git
```

3. Install your local copy into a virtualenv. Assuming you have virtualenvwrapper installed, this is how you set up your fork for local development:

```
$ mkvirtualenv msg_parser
$ cd msg_parser/
$ python setup.py develop
```

4. Create a branch for local development:

```
$ git checkout -b name-of-your-bugfix-or-feature
```

Now you can make your changes locally.

5. When you're done making changes, check that your changes pass flake8 and the tests, including testing other Python versions with tox:

```
$ flake8 msg_parser tests
$ python setup.py test or py.test
$ tox
```

To get flake8 and tox, just pip install them into your virtualenv.

6. Commit your changes and push your branch to GitHub:

```
$ git add .
$ git commit -m "Your detailed description of your changes."
$ git push origin name-of-your-bugfix-or-feature
```

7. Submit a pull request through the GitHub website.



## 5.3 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in README.rst.
3. The pull request should work for Python 2.7, 3.4, 3.5 and 3.6, and for PyPy. Check [https://travis-ci.org/vikramarsid/msg\\_parser/pull\\_requests](https://travis-ci.org/vikramarsid/msg_parser/pull_requests) and make sure that the tests pass for all supported Python versions.

## 5.4 Tips

To run a subset of tests:

```
$ py.test tests.test_msg_parser
```

## 5.5 Deploying

A reminder for the maintainers on how to deploy. Make sure all your changes are committed (including an entry in HISTORY.rst). Then run:

```
$ bumpversion patch # possible: major / minor / patch
$ git push
$ git push --tags
```

Travis will then deploy to PyPI if tests pass.



### 6.1 Development Lead

- Vikram Arsid <vikramarsid@gmail.com>

### 6.2 Contributors

None yet. Why not be the first?



### 7.1 0.1.0 (2018-10-26)

- Initial commit.

### 7.2 1.0.0 (2018-10-29)

- Initial pypi release.
- Added headers\_dict to message object.

### 7.3 1.2.0 (2019-12-12)

- Python 3.x support
- Updates to Makefile
- Formatting and linting



## CHAPTER 8

---

### Indices and tables

---

- `genindex`
- `modindex`
- `search`





### m

- `msg_parser`, 10
- `msg_parser.cli`, 7
- `msg_parser.data_models`, 8
- `msg_parser.email_builder`, 9
- `msg_parser.msg_parser`, 9
- `msg_parser.properties`, 7
- `msg_parser.properties.ms_props_date_type_map`,  
7
- `msg_parser.properties.ms_props_generator`,  
7
- `msg_parser.properties.ms_props_id_map`,  
7



## A

`as_dict()` (*msg\_parser.msg\_parser.Message* method), 9

`Attachment` (class in *msg\_parser.msg\_parser*), 9

## B

`build_email()` (*msg\_parser.email\_builder.EmailFormatter* method), 9

## C

`create_parser()` (in module *msg\_parser.cli*), 8

## D

`DataModel` (class in *msg\_parser.data\_models*), 8

## E

`EmailFormatter` (class in *msg\_parser.email\_builder*), 9

## F

`flatten_list()` (in module *msg\_parser.email\_builder*), 9

`format_size()` (in module *msg\_parser.msg\_parser*), 10

`FullPaths` (class in *msg\_parser.cli*), 7

## G

`generate_data_id_type_mapping()` (in module *msg\_parser.properties.ms\_props\_generator*), 7

`generate_id_name_mapping()` (in module *msg\_parser.properties.ms\_props\_generator*), 7

`generate_master_properties()` (in module *msg\_parser.properties.ms\_props\_generator*), 7

`get_email_mime_content()` (*msg\_parser.msg\_parser.MsOxMessage* method), 9

`get_floating_time()` (in module *msg\_parser.data\_models*), 9

`get_message_as_json()` (*msg\_parser.msg\_parser.MsOxMessage* method), 9

`get_multi_value_offsets()` (in module *msg\_parser.data\_models*), 9

`get_properties()` (*msg\_parser.msg\_parser.MsOxMessage* method), 9

`get_properties_as_dict()` (*msg\_parser.msg\_parser.MsOxMessage* method), 9

`get_time()` (in module *msg\_parser.data\_models*), 9

`get_value()` (*msg\_parser.data\_models.DataModel* method), 9

## I

`is_dir()` (in module *msg\_parser.cli*), 8

`is_valid_msg_file()` (*msg\_parser.msg\_parser.MsOxMessage* method), 9

## L

`lookup_data_type_name()` (*msg\_parser.data\_models.DataModel* static method), 9

## M

`main()` (in module *msg\_parser.cli*), 8

`Message` (class in *msg\_parser.msg\_parser*), 9

`msg_parser` (module), 10

`msg_parser.cli` (module), 7

`msg_parser.data_models` (module), 8

`msg_parser.email_builder` (module), 9

`msg_parser.msg_parser` (module), 9

`msg_parser.properties` (module), 7

`msg_parser.properties.ms_props_date_type_map` (module), 7

`msg_parser.properties.ms_props_generator` (module), 7

`msg_parser.properties.ms_props_id_map` (module), 7

MsOxMessage (class in msg\_parser.msg\_parser), 9

## N

normalize() (in module msg\_parser.email\_builder), 9

## P

parse\_email\_headers() (in module msg\_parser.msg\_parser), 10

PtypBinary() (msg\_parser.data\_models.DataModel static method), 8

PtypBoolean() (msg\_parser.data\_models.DataModel static method), 8

PtypCurrency() (msg\_parser.data\_models.DataModel static method), 8

PtypErrorCode() (msg\_parser.data\_models.DataModel static method), 8

PtypFloating32() (msg\_parser.data\_models.DataModel static method), 8

PtypFloating64() (msg\_parser.data\_models.DataModel static method), 8

PtypFloatingTime() (msg\_parser.data\_models.DataModel static method), 8

PtypGuid() (msg\_parser.data\_models.DataModel static method), 8

PtypInteger16() (msg\_parser.data\_models.DataModel static method), 8

PtypInteger32() (msg\_parser.data\_models.DataModel static method), 8

PtypInteger64() (msg\_parser.data\_models.DataModel static method), 8

PtypMultipleBinary() (msg\_parser.data\_models.DataModel static method), 8

PtypMultipleCurrency() (msg\_parser.data\_models.DataModel static method), 8

PtypMultipleFloating32() (msg\_parser.data\_models.DataModel static method), 8

PtypMultipleFloating64() (msg\_parser.data\_models.DataModel static method), 8

PtypMultipleFloatingTime() (msg\_parser.data\_models.DataModel static method), 8

PtypMultipleGuid() (msg\_parser.data\_models.DataModel static method), 8

PtypMultipleInteger16() (msg\_parser.data\_models.DataModel static method), 8

PtypMultipleInteger32() (msg\_parser.data\_models.DataModel static

method), 8

PtypMultipleInteger64() (msg\_parser.data\_models.DataModel static method), 8

PtypMultipleString() (msg\_parser.data\_models.DataModel static method), 8

PtypMultipleString8() (msg\_parser.data\_models.DataModel static method), 8

PtypMultipleTime() (msg\_parser.data\_models.DataModel static method), 8

PtypNull() (msg\_parser.data\_models.DataModel static method), 8

PtypObject() (msg\_parser.data\_models.DataModel static method), 8

PtypRestriction() (msg\_parser.data\_models.DataModel static method), 8

PtypRuleAction() (msg\_parser.data\_models.DataModel static method), 9

PtypServerId() (msg\_parser.data\_models.DataModel static method), 9

PtypString() (msg\_parser.data\_models.DataModel static method), 9

PtypString8() (msg\_parser.data\_models.DataModel static method), 9

PtypTime() (msg\_parser.data\_models.DataModel static method), 9

PtypUnspecified() (msg\_parser.data\_models.DataModel static method), 9

## R

Recipient (class in msg\_parser.msg\_parser), 10

## S

save\_email\_file() (msg\_parser.msg\_parser.MsOxMessage method), 10

save\_file() (msg\_parser.email\_builder.EmailFormatter method), 9