LDIF

unknown

Oct 04, 2022

CONTENTS

CHAPTER

ONE

LDIF - PARSE AND GENERATE LDIF DATA (SEE RFC 2849).

This is a fork of the `ldif` module from python-ldap with python3/unicode support.

One of its benefits is that it's a pure-python package (you don't depend on the `libldap2-dev` (or similar) package that needs to be installed on your laptop / test machine / production server).

See the first entry in changelog below for a more complete list of differences.

This package only support Python 3 (>= 3.7, actually).

1.1 Usage

Parse LDIF from a file (or `BytesIO`):

```
from ldif import LDIFParser
from pprint import pprint

parser = LDIFParser(open("data.ldif", "rb"))
for dn, record in parser.parse():
    print('got entry record: %s' % dn)
    pprint(record)
```

Write LDIF to a file (or `BytesIO`):

```
from ldif import LDIFWriter

writer = LDIFWriter(open("data.ldif", "wb"))
writer.unparse("mail=alice@example.com", {
    "cn": ["Alice Alison"],
    "mail": ["alice@example.com"],
    "objectclass": ["top", "person"],
})
```

1.2 Unicode support

The stream object that is passed to parser or writer must be an ascii byte stream.

The spec allows to include arbitrary data in base64 encoding or via URL. There is no way of knowing the encoding of this data. To handle this, there are two modes:

By default, the `LDIFParser` will try to interpret all values as UTF-8 and leave only the ones that fail to decode as bytes. But you can also pass an `encoding` of `None` to the constructor, in which case the parser will not try to do any conversion and return bytes directly.

CHAPTER

TWO

API REFERENCE

ldif - generate and parse LDIF data (see RFC 2849).

class ldif.**LDIFParser**(*input_file*, *ignored_attr_types=()*, *process_url_schemes=()*, *line_sep=b'\n'*, *encoding: Optional[str] = 'utf8'*, *strict=True*)

Read LDIF entry or change records from file object.

Parameters

- **input_file** (*file-like object in binary mode*) – file to read the LDIF input from
- **ignored_attr_types** (*List[string]*) – List of attribute types that will be ignored
- **process_url_schemes** (*List[bytearray]*) – List of URL schemes to process with urllib. An empty list turns off all URL processing and the attribute is ignored completely.
- **line_sep** (*bytearray*) – line separator
- **encoding** (*string*) – Encoding to use for converting values to unicode strings. If decoding failes, the raw bytestring will be used instead. You can also pass `None` which will skip decoding and always produce bytestrings. Note that this only applies to entry values. `dn` and entry keys will always be unicode strings.
- **strict** (*boolean*) – If set to `False`, recoverable parse errors will produce log warnings rather than exceptions.

byte_counter

number of bytes that have been read

line_counter

number of lines that have been read

parse() → Iterator[Tuple[Optional[str], dict]]

Iterate LDIF entry records.

Return type
Iterator[Tuple[string, Dict]]

Returns
(dn, entry)

records_read

number of records that have been read

class ldif.**LDIFWriter**(*output_file*, *base64_attrs=()*, *cols=76*, *line_sep=b'\n'*, *encoding='utf8'*)

Write LDIF entry or change records to file object.

Parameters

- **output_file** (*file-like object in binary mode*) – File for output
- **base64_attrs** (*List[string]*) – List of attribute types to be base64-encoded in any case
- **cols** (*int*) – Specifies how many columns a line may have before it is folded into many lines
- **line_sep** (*bytearray*) – line separator
- **encoding** (*string*) – Encoding to use for converting values to bytes. Note that the spec requires the dn field to be UTF-8 encoded, so it does not really make sense to use anything else. Default: `'utf8'`.

records_written

number of records that have been written

unparse(*dn: str*, *record: Union[list, dict]*)

Write an entry or change record to the output file.

Parameters

- **dn** (*string*) – distinguished name
- **record** (*Union[Dict[string, List[string]], List[Tuple]]*) – Either a dictionary holding an entry or a list of additions (2-tuple) or modifications (3-tuple).

CHAPTER

THREE

CHANGELOG

3.1 4.1.2 (2021-10-27)

- Update for Python 3.10.

3.2 4.1.1 (2021-03-04)

- Fix documentation generation -> https://ldif.readthedocs.io/.

3.3 4.1.0 (2021-02-16)

- Add type annotations.

3.4 4.0.0 (2019-11-18)

- Take over the maintenance of this package as we need if for our customers (see: https://github.com/abilian/labandco).
- Drop Python 2 support.

3.5 3.2.2 (2017-02-07)

- Fix detection of unsafe strings in `unparse` (See #7)

3.6 3.2.1 (2016-12-27)

- Ignore non-unicode characters in "dn" in non-strict mode. (Fixes #5)

3.7 3.2.0 (2016-06-03)

- Overhaule the unicode support to also support binary data (e.g. images) encoded in LDIF.

  You can now pass an encoding to the parser which will be used to decode values. If decoding failes, a bytestring will be returned. If you pass an encoding of `None`, the parser will not try to do any conversion and return bytes directly.

  This change should be completely backwards compatible, as the parser now gracefully handles a case where it crashed previously.

  (See #4)

3.8 3.1.1 (2015-09-20)

- Allow empty values for attributes.

3.9 3.1.0 (2015-07-09)

This is mostly a reaction to python-ldap 2.4.20.

- Restore support for `records_read` as well as adding `line_counter` and `byte_counter` that were introduced in python-ldap 2.4.20.
- Stricter order checking of `dn:`.
- Remove partial support for parsing change records. A more complete implementation based on improvements made in python-ldap may be included later. But for now, I don't have the time.

  Breaking change: `LDIFParser.parse()` now yields `dn, entry` rather than `dn, changetype, entry`.

3.10 3.0.2 (2015-06-22)

- Include documentation source and changelog in source distribution. (Thanks to Michael Fladischer)
- Add LICENSE file

3.11 3.0.1 (2015-05-22)

- Use OrderedDict for entries.

3.12 3.0.0 (2015-05-22)

This is the first version of a fork of the `ldif` module from python-ldap. For any changes before that, see the documentation over there. The last version before the fork was 2.4.15.

The changes introduced with this version are:

- Dropped support for python < 2.7.
- Added support for python 3, including unicode support.
- All deprecated functions (`CreateLDIF`, `ParseLDIF`) were removed.
- `LDIFCopy` and `LDIFRecordList` were removed.
- `LDIFParser.handle()` was removed. Instead, `LDIFParser.parse()` yields the records.
- `LDIFParser` has now a `strict` option that defaults to `True` for backwards-compatibility. If set to `False`, recoverable parse errors will produce log warnings rather than exceptions.

PYTHON MODULE INDEX

INDEX