
Mastercard file utilities Documentation

Release 0.6.0+0.g35d491f.dirty

Anthony Delosa

Oct 31, 2018

Contents

1	MasterCard file utilities	3
1.1	Features	3
2	Installation	5
3	Usage	7
3.1	paramconv - Convert MPE parameter extract files	7
3.2	mideu - MasterCard IPM file tool	8
3.3	MasterCard file formats	10
3.4	mciutil package	11
4	Contributing	13
4.1	Types of Contributions	13
4.2	Get Started!	14
4.3	Pull Request Guidelines	15
4.4	Tips	15
5	Credits	17
5.1	Development Lead	17
5.2	Contributors	17
6	History	19
6.1	0.6.0 (2018-10-01)	19
6.2	0.5.0 (2016-10-03)	19
6.3	0.4.8 (2016-10-02)	19
6.4	0.4.6 (2016-08-09)	19
6.5	0.4.5 (2016-08-06)	20
6.6	0.4.4-0.4.3 (2016-08-03)	20
6.7	0.4.2 (2016-03-13)	20
6.8	0.4.1 (2015-12-16)	20
6.9	0.4.0 (2015-10-05)	20
6.10	0.3.0 (2015-10-03)	20
6.11	0.2.0 (2015-08-28)	20
6.12	0.1.0 (2015-08-20)	21
7	Indices and tables	23

Contents:

MasterCard file utilities

Set of command line utilities to work with various MasterCard files.

- Free software: BSD license
- Documentation: <https://mciutil.readthedocs.org>.

1.1 Features

Provides the following command line utilities:

- paramconv: Utility for working with MasterCard MPE parameter extract files
- mideu: Utility for working with MasterCard IPM files

CHAPTER 2

Installation

mcutil is distributed as a python package.

You will need a version of python installed to run the tool.

Python version 2.7 or 3.4 and above are supported. This will most likely work on Python 2.6 but no testing is done and no 2.6 fixes will be considered. Its time to move to Py3.

See <https://www.python.org/downloads> for instructions on installing python on your platform of choice.

Once you have Python available, at the command line enter:

```
$ virtualenv mciutil
$ source ./mciutil/bin/activate
$ pip install mciutil
```

For Windows platforms:

```
c:> virtualenv mciutil
c:> ./mciutil/Scripts/activate
c:> pip install mciutil
```

If you start another command prompt after installing, you can make the commands available in the new session by activating the python environment:

```
$ source ./mciutil/bin/activate
```

or for windows:

```
c:> ./mciutil/Scripts/activate
```

Usage

The mciutil package provides command line tools and python functions you can use when working with MasterCard files.

Note: Make sure set binary transfers when moving MasterCard files. For example, if using FTP, use the `binary` command before starting the transfer

3.1 paramconv - Convert MPE parameter extract files

Use this tool if you are moving between a mainframe and PC based clearing application and need the parameter files available on both platforms. The app currently works with 1014 blocked file and VBS format which is used for communications between MasterCard and the member.

Most simple usage, just provide a MasterCard MPE file to convert:

```
paramconv <inputfile>
```

This runs with the following assumptions

- Input file format is EBCDIC with 1014 blocking
- Output file name is the input file name plus a '.out' extension

If you have an ASCII file and want to convert it to EBCDIC, you need to provide the source format type:

```
paramconv -s ascii <inputfile>
```

If you are working with VBS format, just add the `--no1014blocking` flag to the command:

```
paramconv -s ascii --no1014blocking <inputfile>
```

You can change the output file location:

```
paramconv -o <outputfile> <inputfile>
```

To get all the usage details:

```
paramconv --help
```

3.2 mideu - MasterCard IPM file tool

This utility is for working with MasterCard IPM clearing files. Currently the utility provides the following functions:

- extract - provide IPM transaction data in usable format
- convert - convert IPM files between ASCII and EBCDIC encoding

3.2.1 Extract command

Use this command to extract transactions from a MasterCard IPM format file into usable formats like csv. The app currently works with 1014 blocked file format and VBS format which is used for communications between MasterCard and the member.

Get a csv file from an IPM file:

```
mideu extract <filename>
```

This runs with the following assumptions

- Input file is EBCDIC and 1014 blocked format
- Output file name is the input file name plus a '.csv' extension

Attention: Currently python 2.6 does not print a header row as the csv library only added this support in python 2.7. This function may be added in a future release.

If you need to process an ASCII encoded 1014 blocked file:

```
mideu extract -s ascii <filename>
```

If you are working with VBS format, just add the `--no1014blocking` flag to the command:

```
mideu extract -s ascii --no1014blocking <filename>
```

You can change the CSV output file name and location:

```
mideu extract <inputfile> --csvoutputfile <outputfile>
```

To get all the usage details:

```
mideu extract --help
```

3.2.2 Convert command

Use this command to convert a MasterCard IPM format file into another encoding scheme (ASCII or EBCDIC). The app currently works exclusively with 1014 blocked file format which is used for communications between MasterCard and the member.

Most simple usage, just provide a MasterCard IPM file to convert:

```
mideu convert <inputfile>
```

This runs with the following assumptions

- Input file is EBCDIC and 1014 blocked format
- Output file name is the input file name plus a '.out' extension

If you have a ASCII file and want to convert it to EBCDIC, you need to provide the source format type:

```
mideu convert -s ascii <inputfile>
```

If you are working with VBS format, just add the `--no1014blocking` flag to the command:

```
mideu convert -s ascii --no1014blocking <filename>
```

To get all the usage details:

```
mideu convert --help
```

3.2.3 mideu.yml configuration

The package provides a common configuration, but you can provide your own if you wish to change it. You have 2 options for providing your own configuration.

- mideu.yml file in the current directory
- .mideu.yml file in the users home directory

The file is a basic yaml file with the following sections

output_data_elements Specify fields to output. Set the order and fields to be output.

Structure:

```
output_data_elements:
  - MTI
  - DE1
  - DE10
  - PDS0023
  - DE43_NAME
```

bit_config Define the bitmap fields. You should not need to change this but if you think you need to, have a look at the source to see what the options mean. Will document this in more details at some point.

Structure:

```
bit_config:
  1:
    field_name: Bitmap secondary
    field_type: FIXED
```

(continues on next page)

(continued from previous page)

```

    field_length: 8
2:
    field_name: PAN
    field_type: LLVAR
    field_length: 0
    field_processor: PAN

```

3.3 MasterCard file formats

3.3.1 VBS file format

added 0.4.6

This format is a basic variable record format. To process this format, add the `--no1014blocking` option.

There are no carriage returns or line feeds in the file. A file consists of records. Each record is prefixed with a 4 byte binary length.

Say you had a file with the following 2 records:

```

"This is first record 1234567" <- length 28
"This is second record AAAABBBBBB123" <- length 34

```

Add binary length to the start of each record. (x'1C' = 28, x'22' = 34) with the file finishing with a zero length record length:

```

00000000: 00 00 00 1C 54 68 69 73 20 69 73 20 66 69 72 73 ....This is fir
00000010: 74 20 72 65 63 6F 72 64 20 31 32 33 34 35 36 37 t record 1234567
00000020: 00 00 00 22 54 68 69 73 20 69 73 20 73 65 63 6F ..."This is seco
00000030: 6E 64 20 72 65 63 6F 72 64 20 41 41 41 41 42 42 nd record AAAABB
00000040: 42 42 42 31 32 33 00 00 00 00 BBB123....

```

3.3.2 1014 blocked file format

This is the default format used by mciutil

This is the same as VBS format with 1014 blocking applied.

The VBS data is blocked into lengths of 1012, and an additional 2 x'40' characters are appended at each block.

Finally, the total file length is made a multiple of 1014 with the final incomplete record being filled with the x'40' character

Taking the above VBS example

```

00000000: 00 00 00 1C 54 68 69 73 20 69 73 20 66 69 72 73 ....This is fir
00000010: 74 20 72 65 63 6F 72 64 20 31 32 33 34 35 36 37 t record 1234567
00000020: 00 00 00 22 54 68 69 73 20 69 73 20 73 65 63 6F ..."This is seco
00000030: 6E 64 20 72 65 63 6F 72 64 20 41 41 41 41 42 42 nd record AAAABB
00000040: 42 42 42 31 32 33 00 00 00 00 BBB123....

```

Block to 1014 by adding 2 * x'40' characters every 1012 characters in the data. Finally fill with x'40' characters to next 1014 increment. In this case, there is only one increment:

```

00000000: 00 00 00 1C 54 68 69 73 20 69 73 20 66 69 72 73 ....This is firs
00000010: 74 20 72 65 63 6F 72 64 20 31 32 33 34 35 36 37 t record 1234567
00000020: 00 00 00 22 54 68 69 73 20 69 73 20 73 65 63 6F ..."This is seco
00000030: 6E 64 20 72 65 63 6F 72 64 20 41 41 41 41 42 42 nd record AAAABB
00000040: 42 42 42 31 32 33 00 00 00 00 40 40 40 40 40 40 BBB123....@@@@@@
00000050: 40 40 40 40 40 40 40 40 40 40 40 40 40 40 40 40 @@@@@@@@@@@@@@@@@@@@
... all X'40' characters
000003E0: 40 40 40 40 40 40 40 40 40 40 40 40 40 40 40 40 @@@@@@@@@@@@@@@@@@@@
000003F0: 40 40 40 40 40 40 40 40 40 40 40 40 40 40 40 40 @@@@@@@

```

3.4 mciutil package

To use Mastercard file utilities in a project:

```
import mciutil
```

There are some useful functions for working with bitmap, variable length files. Will document in a future version.

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

4.1 Types of Contributions

4.1.1 Report Bugs

Report bugs at <https://github.com/adelosa/mciutil/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.

4.1.2 Fix Bugs

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

4.1.3 Implement Features

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

4.1.4 Write Documentation

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

4.1.5 Submit Feedback

The best way to send feedback is to file an issue at <https://github.com/adelosa/mciutil/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 :)

4.2 Get Started!

Ready to contribute? Here's how to set up *mciutil* for local development.

1. Fork the *mciutil* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/mciutil.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 mciutil
$ cd mciutil/
$ 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 mciutil tests
$ python setup.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.

4.3 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, and 3.5, and for PyPy. Check https://travis-ci.org/adelosa/mciutil/pull_requests and make sure that the tests pass for all supported Python versions.

4.4 Tips

To run a subset of tests:

```
$ python -m unittest tests.test_mciutil
```


5.1 Development Lead

- Anthony Delosa <adelosa@gmail.com>

5.2 Contributors

None yet. Why not be the first?

6.1 0.6.0 (2018-10-01)

- Removed dependency on bitarray (no binary wheels)
- Added details on installation for non-python users
- 2 years almost since last update!

6.2 0.5.0 (2016-10-03)

- Fixed version display in release version.
- Removed support for mongo extract

6.3 0.4.8 (2016-10-02)

- added support for latin1 encoding of csv extract
- fixes to setup process so that mideu.yml file is installed
- fixed de43 split to allow more formats for different countries

6.4 0.4.6 (2016-08-09)

- added `--no1014blocking` option to allow processing of VBS structure files.

6.5 0.4.5 (2016-08-06)

- check that all of message consumed by fields otherwise raise exception
- get rid of a heap of debugging prints that were clogging the output
- allow freestyle de43 fields with the de43 processor enabled. Use regex rather than string splits

6.6 0.4.4-0.4.3 (2016-08-03)

- Fix issue with mideu when no parameters passed (stack trace)
- Some more debugging messages provided with -d switch
- signed the release with key for 0.4.4. need to publish my pub key somewhere..

6.7 0.4.2 (2016-03-13)

- Complete data elements added to default config.
- Added versioneer support for easier package versions

6.8 0.4.1 (2015-12-16)

- Additional data elements added to default config file.

6.9 0.4.0 (2015-10-05)

- Now supporting python 2.6 (for all those still using RHEL 6)
- Headers rows in mideu csv extracts don't work in 2.6

6.10 0.3.0 (2015-10-03)

- added sub commands for mideu
- mideu now supports IPM encoding conversion between ascii and ebcdic
- Now faster using list comps instead of slow loops

6.11 0.2.0 (2015-08-28)

- Support for config override for mideu - see usage doco
- Progress bar while using mideu.. it takes a while
- Now supports python 3.4, 3.5 and 2.7. Upgrade if you are using 2.6
- New usage documentation

6.12 0.1.0 (2015-08-20)

- First release.

CHAPTER 7

Indices and tables

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