



GNU Recutils

<http://www.gnu.org/software/recutils>

August 24, 2013



Outline

- 1 Introduction
- 2 The Format
- 3 The Software
- 4 New Features in 1.6
- 5 Ideas for the future



GNU Recutils

Set of tools and libraries to access **human-editable, plain text databases** called recfiles.



The Format

```

      .-""""-.
      |"   (a \
      \--   |
      ;,  --.;.
      / '""""'\#'.
      ._|           \##\
      ( |           /'###;
      / 0o '\      .-""":      :'"-.      .-' |##|
      |__ - | ,'      .      '      ', ( _.' |##|
      .->   \ /:      :      ;      : ' ' /##/
      '-.    '\'.      .      :      ' \   .##'
      ' .    | .'. _.' ' _.' ' _.' . ;--;##;'
      '-\ .   :      :      '      ' / | \ (
      .-' ' _ '      .      : _ . \ (
      ((((-) / '";--...:..--;"' \ / \
      . ' /           \ ( _ _ _ . '
      ((((-)         ((((-)

```



Fields and Records

Motiv: Windows 7 Sins

Type: Sticker

Amount: 1

Motiv: iBad

Type: Sticker

Type: Roll

Amount: 1

Motiv: GPLv3

Type: Sticker

Amount: 1

Note: Need more.



Field Values

- Simple form:

Email: jemarch@gnu.org

- Splitting logical lines:

LongLine: This is a quite long value \\
composed by a unique logical line \\
splitted in several physical lines.

- Continuation lines:

Address: DonCojon GmbH
+ Einbahnstrasse 100
+ 60231 Ausfahrt am Main
+ Germany



Comments

```
# -*- mode: rec -*-  
#  
# This file contains a list of contacts.  
  
Name: Mr. Foo  
Email: foo@foo.com  
  
Name: Mr. Bar  
# Not valid since 2009:  
# Email: bar@bar.com  
Email: altbar@bar.com  
  
# End of contacts.rec
```



Record Descriptors

%rec: Maintainer

Name: John Thompson

Email: john@thompson.com

%rec: Package

Name: GNU Foo

URL: <http://www.gnu.org/software/foo>

Name: GNU Bar

URL: <http://www.gnu.org/software/bar>



%-fields

- %rec: TYPE
- %mandatory, %prohibit, %unique
- %key, %doc
- %type, %typedef
- %sort, %auto, %size, %confidential
- %constraint



gnu.rec with data integrity

```
%rec: Maintainer  
%key: Name  
%type: Email email
```

```
Name: John Thompson  
Email: john@thompson.com
```

```
%rec: Package  
%key: Name  
%mandatory: URL
```

```
Name: GNU Foo  
URL: http://www.gnu.org/software/foo
```

```
Name: GNU Bar  
URL: http://www.gnu.org/software/bar
```



Field types: %type and %typedef

- Simple types:

```
%type: FNAME (int|bool|real|line|date|email)
```

- Strings with maximum size:

```
%type: FNAME size NUMBER
```

- Regular expressions:

```
%type: FNAME regexp /REGEXP/
```

- Enumerations:

```
%type: FNAME enum VAL1 VAL2 VAL3
```

- Typedefs:

```
%typedef: code_t int
```



Enumeration type

- An example of an enumeration type:

```
%rec: Task
```

```
%type: Status enum TODO ASSIGNED DONE DISCARDED
```

- Text enclosed in parenthesis is ignored:

```
%rec: Task
```

```
%type: Status enum
```

```
+ TODO      (New task, unassigned)
```

```
+ ASSIGNED  (Someone is working in the task)
```

```
+ DONE      (Task is done)
```

```
+ DISCARDED (Task was discarded)
```



Regex type

- An example of a regular expression type:

```
%rec: Product
```

```
%type: Id regexp /[A-Z][0-9]{4}/
```

```
Id: B0006
```

- Like in sed, any character can delimit the regexp:

```
%rec: File
```

```
%type: Path regexp ,(/[a-zA-Z0-9]+),
```

```
Path: /foo/bar
```



Encryption Support

- Selective: individual fields can be encrypted.
- Password-based AES.

```
%rec: List
%key: Address
%type: Address,SubscribedAs email
%confidential: Password AdminPassword
```

```
Address: platform-testers@gnu.org
Password: encrypted-XXXXXXXXXXXXXXXXXXXXXXXXXX==
AdminPassword: encrypted-XXXXXXXXXXXXXXXXXXXXXXXXXX==
SubscribedAs: jemarch@gnu.org
```

```
Address: cockroach-devel@gnu.org
Password: encrypted-XXXXXXXXXXXXXXXXXXXXXXXXXX==
AdminPassword: encrypted-XXXXXXXXXXXXXXXXXXXXXXXXXX==
SubscribedAs: jemarch@gnu.org
```



Remote Record Descriptors

```
%rec: FSD_Entry http://www.jemarch.net/downloads/FSD.rec
```

```
Title: GNU Recutils
```

```
Description: GNU Recutils is a set of tools ...
```

```
GNU: yes
```

```
Homepage: http://www.gnu.org/software/recutils
```

```
PublicVCSCheckout: git clone git://git.sv.gnu.org/recutils.git
```

```
License: GPLv3PLUS
```

```
License: GFDLv21PLUS
```

```
Maintainer: Jose E. Marchesi <jemarch@gnu.org>
```

```
...
```



Auto Generated Fields

%rec: Item
%key: Id
%type: Created date
%auto: Id, Created

Id: 10
Created: Sun Nov 10 08:20:00 CET 2011
Title: An item

Id: 11
Created: Sun Nov 13 12:31:15 CET 2011
Title: Another item



The Software



Rich C API to manage recfiles.

- Management of fields, records, record sets and databases.
- Management of types.
- Record selection expressions.
- Field resolvers.
- Rec parser.
- Rec writer.
- ...



The Utilities

- **recinf**: Printing information about recfiles.
- **recsel**: Selecting records.
- **recins**: Inserting records.
- **recdel**: Deleting records.
- **reset**: Adding/Setting/Deleting fields in a record.
- **recfix**: Checking data integrity.
- **recfmt**: Formatting records using templates.
- **csv2rec**, **rec2csv**, **mdb2rec**: Converting from/to other formats.



recinf - Printing information about recfiles

Usage: recinf [OPTION]... [FILE]...

Print information about the contents of recfiles.

```
$ recinf gnu.rec
```

```
3 Maintainer
```

```
4 Package
```

```
$ recinf -d -t Package gnu.rec
```

```
%rec: Package
```

```
%key: Name
```



recsel - Selecting records

Usage: recsel [OPTION]... [FILE]...

Select and print rec data.

- Getting records of a **given type**:

```
$ recsel -t Package gnu.rec
```

```
Name: GNU Foo
```

```
URL: http://www.gnu.org/software/foo
```

```
Name: GNU Bar
```

```
URL: http://www.gnu.org/software/bar
```

- Printing the **nth record** of a given type:

```
$ recsel -t Package -n 1 gnu.rec
```

```
Name: GNU Bar
```

```
URL: http://www.gnu.org/software/bar
```



recsel - Selecting records

- Print records satisfying a given **expression**:

```
$ recsel -t Maintainer -e "#Email > 0" gnu.rec  
$ recsel -t Maintainer -e "Email ~ '.org'" gnu.rec  
$ recsel -t Maintainer -e "Email[0] ~ '.org'" gnu.rec  
$ recsel -e "Cyclo > 10 && !Tested" functions.rec
```

- Print a **subset of the fields**:

```
$ recsel -t Maintainer -p Email gnu.rec  
Email: john@thompson.com
```

```
Email: tom@johnson.com
```

- Print **values** instead of fields:

```
$ recsel -t Maintainer -P Email gnu.rec  
john@thompson.com
```

```
tom@johnson.com
```



recsel - Selecting records

- **Collapse** the output:

```
$ recsel -C -t Maintainer -P Email gnu.rec  
john@thompson.com  
tom@johnson.com
```

- Print values in a **row**:

```
$ recsel -C -R Id,Title tasks.rec  
1 First task title.  
2 Second task title.  
3 Third task title.
```



recsel - Selecting records

- Select **encrypted** fields:

```
$ recsel -C -s secret lists.rec
```

```
Address: platform-testers@gnu.org
```

```
Password: apassword
```

```
AdminPassword: anotherpassword
```

```
SubscribedAs: jemarch@gnu.org
```

- **Sort** the output:

```
$ recsel -S Name contacts.rec
```

```
Name: Abraham Abramovich
```

```
Email: aa@foo.org
```

```
Name: Alejandro Brebia
```

```
Email: abrebia@bar.org
```



recins - Inserting records

Usage: recins [OPTION]... [-f STR -v STR]... [FILE]
Insert records in a recfile.

- Altering the contents of a file:

```
$ recins -t Package -f Name -v "GNU foo" \  
        -f URL -v "http://foo.org" gnu.rec
```

- Working as a filter:

```
$ recins -t Package -f Name -v "GNU foo" \  
        -f URL -v "http://foo.org" < gnu.rec  
... existing records ...
```

```
Name: GNU foo  
URL: http://foo.org
```



recdel - Deleting records

Usage: recdel [OPTION]... [FILE]

Remove (or comment out) records from a recfile.

- Remove all records of a given type:

```
$ recdel -t Package gnu.rec
```

- Comment out the 10th contact:

```
$ recdel -n 10 -c contacts.rec
```

- Using a record selection expression:

```
$ recdel -e "Email[0] = 'foo@bar.com'" contacts.rec
```



recset - Adding/Setting/Deleting fields in a record

Usage: recset [OPTION]... [FILE]...
Alter or delete fields in records.

- Adding a field to a given record:

```
$ recset -t Maintainer -e "Email ~ "hotmail.com" \  
-f Note -a "WTF" gnu.rec
```

- Changing the value of a field:

```
$ recset -t Package -n 10 -f URL -s "http://new.url" gnu.rec
```

- Removing/commenting out fields in a record:

```
$ recset -t Maintainer -f Email[1],Email[2] -d  
$ recset -t Maintainer -f Email[1],Email[2] -c
```



recfix - Checking data integrity

Usage: recfix [OPTION]... [OPERATION] [FILE]...

Check and fix rec files.

- Checking integrity of a rec file:

```
$ recfix gnu.rec
```

```
gnu.rec:6: error: expected 'int' value
```

```
gnu.rec:15: error: mandatory field 'URL' not found in record
```

- Physically sort a file:

```
$ recfix --sort gnu.rec
```

- Encrypt unencrypted confidential fields:

```
$ recfix --encrypt -s secret gnu.rec
```

- Decrypt encrypted confidential fields:

```
$ recfix --decrypt -s secret gnu.rec
```



Converters from/to other formats

- CSV to rec
\$ `csv2rec foo.csv > foo.rec`
- rec to CSV
\$ `rec2csv foo.rec > foo.csv`
- mdb to rec (with data integrity)
\$ `mdb2rec foo.mdb > foo.rec`



rec-mode

- Emacs mode to edit rec files.
- Font lock.
- Navigation through records.



ob-rec: Integration with org-babel

```
#+begin_src rec :data gnu.rec :type Maintainer :fields Name,Email
Email ~ '.*com'
#+end_src
```

```
#+results:
```

Name	Email
John Thompson	john@thompson.com
Thomas Johnson	tom@johnson.com



New Features in 1.6



- Recutils 1.6 is **imminent!!!**
- But updating the user manual is sooooo booooring.
- Also, I simply cannot stop adding new features.



New Operators in Selection Expressions

- \geq and \leq
 `recsel -e 'Age \geq 18 && Age \leq 66' persons.rec`
- They were documented, but not implemented! XD



New field type: UUID

`%type: Id uuid`

- Universally unique identifiers.
- Support for uuid auto-fields.



Rewrite Rules for Fields

```
$ recsel -n 2 -p Name,Email:Address contacts.rec  
Name: Mr. Foo Bar  
Address: foo.bar@baz.com
```

- Allow to rename fields in the output data.
- Very useful when combined with **joins** and **aggregates**.



Sorting by Multiple Fields

```
$ recsel -S Date,Type sales.rec
```

- List of comma-separated field names to -S and -sort.
- Lexicographic ordering.



Grouping

- New option `-G/-group-by=FIELD,...` for `recsel`.
- Similar to the SQL “GROUP BY” construction.
- But better defined: it unifies records removing duplicates.
- Example:

```
$ recsel -G Date -p Date,Id sales.rec
```

```
Date: 12 September 2010
```

```
Id: 120
```

```
Id: 121
```

```
Id: 123
```

```
Id: 130
```

```
Date: 13 September 2010
```

```
Id: 140
```

```
Id: 142
```



Aggregate Functions

- Function calls in fexes.
- Case-insensitive names.
- Names of the generated fields: Count(Foo) => Count_Foo
- Fixed predefined set: Count(), Sum(), Avg(), Min(), Max().
- Example:

```
$ recsel -p Name,Avg(Score) -G Name evals.rec  
Name: Jaimito Mueller  
Avg_Score: 8.9
```

```
Name: Pedrito Copon  
Avg_Score: 4.2
```



Joins

- The new type *rec*.

`%rec: Package`

`%key: Name`

`%type: Maintainer rec Hacker`

- New option `-j|-join=FIELD` for `recsel`.
- Dot-notation in selection expressions.
- Dot-notation in `fexes`.

```
$ recsel -j Maintainer -e "Maintainer.Email ~ '.org'" \  
-p Name,Maintainer.email
```

```
Name: GNU recutils
```

```
Maintainer_Email: jemarch@gnu.org
```

```
Name: GNU Epsilon
```

```
Maintainer_Email: positron@gnu.org
```



High-level API in librec

Almost no logic in the tools: everything is in librec.

- `rec_db_query ()`
- `rec_db_insert ()`
- `rec_db_delete ()`
- `rec_db_set ()`



Support for Binary Indexes

- Developed as part of the Google SoC 2012.
- A binary index file is created using `recfix`, containing several indexes.
- `rec_db_query` then loads and uses the index file if it exists.
- Needs to be merged.



Bindings to other Languages

- Scheme (Guile).
- Algol 68.
- Python (GSOC 2013).



A much improved rec-mode

- Demo..



Lots of bug fixes!

- And lots of new bugs as well!



Ideas for the future



recsql

- The recutils command-line interface is very comfortable to use.
- But people is familiar with SQL.
- Simple mapping to the high-level librec API.
- Could include an interactive shell as wel...



Support for Transactions

```
$ rectrans foo.rec (sets REC_TRANS_NO=23000)
                   (creates foo.trans)
```

```
$ recins -f foo -v bar foo.rec
```

```
...
```

```
$ recdel --transaction=23000 -n 2 foo.rec
```

```
...
```

```
$ rectrans --rollback
```

```
$ rectrans -e foo.rec (implies commit)
```



More Powerful Templates

- Improved language supported by recfmt.
- Conditionals.
- Comments.
- Loops.
- Inclusion of other files.
- Output splitted in several files.



Multi-valued enum fields

- This is verbose:

Id: 102

Title: The timing of load instructions is wrong.

Entity: ERC32

Entity: LEON2

Entity: LEON3

- Consider this:

Id: 102

Title: The timing of load instructions is wrong.

Entity: ERC32 LEON2 LEON3

- Commas are ignored:

Entity: ERC32, LEON2, LEON3



Bash goes recutils!

Loadable bash builtins.

```
ls --rec | while readrec
do
    if [% LastModified << 12/08/2013 %]; then
        touch 'echo "$REPLY_REC" | recsel -P FileName'
    fi
done
```



Compatibility with Debian Files

- Debian (and derived distros) uses text files to maintain the packages database.
- The format is **almost** a subset of the rec format, but:
 - Different strategy for continuation lines.

Description-en: The GNU Emacs editor (metapackage)

GNU Emacs is the extensible self-documenting text editor

This is a metapackage which will always depend on the latest
release.

.

Other paragraph.

- Extensive usage of multi-valued fields.
Tag: devel::editor, role::dummy, ...
- Shall we be compatible? How?



Thanks for Listening

<http://www.gnu.org/software/recutils>



Other ideas

- Sed-like editing in recset:
`$ recset -t Contact -f Email[1] -E 's/.com/.org/'`
- Scalar functions in selection expressions:
`$ recsel -e 'Age > Avg(10,50)'`
- User-defined set of aggregates and scalar functions (guile).
- Utils for merging and splitting record sets.

