Records-Mode

Personal diary software
Edition 1.0, for Records version 1.4
January 2000

by Ashvin Goel

1 Introduction

The *records-mode* for emacs allows creation of an online diary. Users can create records in their diary for different subjects each day. Records provides linking and indexing facilities so that users can easily collect their thoughts on a subject over a period of time. For instance, it is easy to find out whether you were thinking of quitting school two months back! Records provides a reminder service by collecting your list of 'to do' things. A simple integration with the emacs calendar is also available.

2 Where to get records

You can get the latest version of records from `http://www.cse.ogi.edu/~ashvin/software.html`. Soon this project will be migrating to `http://sourceforge.net`. Then the latest version will be available from `http://sourceforge.net/projects/records`.

3 Installation

The steps below will help you install records. The installation of records is also documented in the README and the INSTALL file available with the distribution. The number of steps in the install (or the amount of text below) seems to be ungodly! Panic not, almost everything in the install is automated. Much of the text below is only explaining what the install is doing (or not doing) to your computer!

The install procedure makes the byte compiled elisp files, reads in the user settings for records and installs these settings into the records initialization file '`~/.emacs-records`'. This initialization file is read by emacs and the records administration program ('`recordsadmin`'). Install also adds the loading of '`~/.emacs-records`' into your '`~/.emacs`'. When you start emacs, all the records defaults will be automatically loaded.

1. Since you are reading this file, you have untarred the gzip'ed file. The elisp files in the current directory (in which you are right now) will *not* be installed in any other directory, and installation will automatically add the current directory to the load path in emacs. So if you wish to move the records software to some other emacs-related directory, now is the time to do it.
2. If you are installing records on Windows, read INSTALL-NT first.
3. Run `./configure` or `./configure --enable-xemacs`

   The `--enable-xemacs` flag ensures that the elisp files are compiled using `xemacs`. You MUST use this option if you use `xemacs`.

   The configure is provided in the source directory (do not run your own configure). If you want to specify the directory in which '`recordsadmin`' (a Perl script) should be installed, you should run

   `./configure --prefix $HOME`

   Make sure `${bindir}` (generally `${prefix}/bin`) in the Makefile exists and is writable by you.

Note: if you are getting source from CVS, you will have to generate `./configure` by running `autoconf` first.

4. Run `make install` (or just `make`)

   Make compiles the lisp files and then interactively asks for the user defaults. Note that the lisp files are not installed anywhere.

5. The changes to your system that `make install` does are the following:
   1. Adds '`~/.emacs-records`' (if this file already exists, it converts only specific parts of the file).
   2. Adds a couple of lines to '`~/.emacs`'
   3. Adds '`recordsadmin`' to `${bindir}` (if you invoked `make install`).
   4. Creates your records directory ('`~/records`' or whatever you have specified during `make`) and adds some indexes there. If this directory already exists, then it re-indexes your records and converts them to the new format. If you don't trust the conversion, I would suggest that you KEEP A COPY OF YOUR RECORDS BEFORE DOING THE CONVERSION. Check the conversion by hand to make sure that the conversion is fine. Specially look for the following: HAVE EMBEDDED LINKS IN YOUR RECORDS BEEN CONVERTED CORRECTLY. Please tell me if this conversion is not okay.

6. Copy the next set of lines from '`;;;; records-mode`' to '`;;;; records-mode end`' into your '`~/.emacs`'. Make sure that these lines occur after the following line (which has been automatically added during install) in your '`.emacs`'.

   ```
   (load "~/.emacs-records")
   ```

   I did not automate this process since users may wish to change the key settings. Users will probably use the records-goto-today function most often. Bind the function to a simple key.

   ```
   ;;;; records-mode
   ; Define key bindings for functions called from outside records mode

   ; The preferred binding for records-goto-today - uncomment to use it
   (define-key global-map [?\C-c ?n] 'records-goto-today)
   (define-key global-map [?\C-c ?-] 'records-underline-line)

   ; The Ctrl-x n map
   (define-key global-map [?\C-x ?n ?t] 'records-goto-today)
   (define-key global-map [?\C-x ?n ?r] 'records-insert-record-region)
   (define-key global-map [?\C-x ?n ?b] 'records-insert-record-buffer)
   (define-key global-map [?\C-x ?n ?l] 'records-insert-link)

   ; Hook up to the calendar mode
   (add-hook 'calendar-load-hook
     (function
      (lambda ()
        (define-key calendar-mode-map "n"
                 'records-calendar-to-record))))
   ```

```
; Hook up to the latex mode
(add-hook 'LaTeX-mode-hook
          (function
           (lambda ()
             (define-key LaTeX-mode-map "\C-clw" 'records-widen-latex))))

; Hook to insert a template when a record is inserted.
; See function records-insert-template
(add-hook 'records-make-record-hook
          (function (lambda ()
                      (records-insert-template current-prefix-arg))))
;;;*** OPTIONAL ***;;;

; If you like abbrev mode
(add-hook 'records-mode-hooks
    (function
     (lambda ()
       (abbrev-mode 1))))

; If you like to use auto filling and/or filladapt modes while editing text.
; If you have these functions in the text-mode-hook, then you don't need
; them here.
(add-hook 'records-mode-hooks
      (function (lambda ()
                  (turn-on-auto-fill)
                  (turn-on-filladapt-mode)
                  )))

; If you want to always start up in outline mode
(add-hook 'records-mode-hooks 'records-outline-mode)

; Using imenu: imenu will invoke a popup menu of records subjects
; 1. In Xemacs:
; (define-key global-map [(shift button3)] 'imenu)
; 2. In gnu emacs:
; (define-key global-map [(shift mouse-3)] 'imenu)

; If you want to be brought to today's record on startup
(records-goto-today)

;;;; records-mode end
```

7. The installation is complete. Restart your emacs (ya, I know that you hate to restart emacs because it is your OS. Well you don't have to restart emacs if you know how to evaluate regions in a buffer. In this case, evaluate the key bindings and all the hooks defined above and load in '`~/.emacs-records`'). Now that the installation is complete, you are ready to start using records. First, read the short tutorial (see Chapter 4 [Tutorial], page 4) on getting started.

4 A Real Quick Tutorial

Here is a real short tutorial on getting started with records.

1. Type key *C-c n* (for `records-goto-today`). This will get you today's record file. If you are starting fresh, it will be empty.
2. A record file consists of records, each of which has a *subject* and a *body*. To insert a record, type *C-c C-i* (`records-insert-record`) or look for "Insert Record" in the menu. Type in a subject in the modeline and press return. The record subject will be added and you can start typing in the record body. You can have spaces in subject titles. Look at the FAQ in the distribution.
3. Multiple records with different subjects can be added in each day's record file. In fact the same subject can be added multiple times in each day's record file.
4. For much of the other functionality, look at the records menu. The menu has been divided so that most of the functionality at the top of the menu is related to records traversal (up, down, prev, next, goto etc). The middle section adds, deletes and renames records. The last section does administration tasks such as encryption, concatenation of records by subject or days, etc. Make sure you look at the records TODO facility (see Section 6.1 [Organizing your TODOs], page 9).
5. Remember that if you just have one record file (today's) then records traversal's are not very useful!
6. The program '`recordsadmin`' has been provided in the distribution in order to administer your records. You should not need to use '`recordsadmin`' in normal records use. '`recordsadmin`' 1) performs initialization (or re-initialization) of your records software, 2) allows changing the date format, 3) changing the directory structure, and 4) recreation of records indexes if they are broken (for example if your system crashes while emacs is updating your indexes).

5 Basic Use

This section describes the basic functionality of the records mode.

5.1 What is a Record?

A record is a thought, or an idea, or even a manuscript with a name. In records-mode, the name of a record is called the *record subject*. The contents of the record are called the *record body*. Here, within double quotes, is an example of a record.

```
"
* Work
------
link: <../../99/04/042199#* Work>

Didn't do my experiments because I continued working on the records
documentation. Hope to start experiments tomorrow.
```

"

The records subject is `Work`. A subject starts with an asterisk and a space and is always underlined. Below the underlining, a link identifies the record. The link above shows that this record is associated with the date April 21st, 1999. Note that there are many ways in which dates can be represented in records (see Section 5.7 [Records Administration], page 8).

If the variable `records-use-font-lock` is set to true (the default), the subject, records link and the TODO (see Section 6.1 [Organizing your TODOs], page 9) are fontified. You can modify the fontification by changing the variable `records-mode-font-lock-keywords`. If the variable `records-subject-read-only` is set to true (the default), the subject is read-only, which disallows any accidental updates to a records subject.

5.2 What is a Record File?

A record file consists of a set of records associated with a given date. For instance, a record file for April 21st, 1999 might contain a `Movies` record and two `Work` records. Simple isn't it!

5.3 Record Traversal

One of the promises made up-front about the records-mode is that it allows easy linking and indexing of your personal ideas and thoughts. Records allows traversing records within a record file and across files.

`records-goto-today`
: Key *C-c n*. Go to the records file of today. This is probably the first function you will be using with records.

`records-goto-up-record`
: Key *C-M-a*. Move to the previous record in the file.

`records-goto-down-record`
: Key *C-M-e*. Move to the next record in the file.

`records-goto-prev-record`
: Key *C-c C-p*. Move to the previous record with the same subject. A previous same-subject record can be on the same date (when there are two or more records with the same subject on the same date) or it is associated with a previous date. Think of all the records on a given subject as a sequence sorted by date. This function moves back one step in this sequence.

`records-goto-next-record`
: Key *C-c C-n*. Move to the next record with the same subject. See `records-goto-prev-record` for more details.

`records-goto-prev-record-file`
: Key *C-c C-b*. Go to the previous record file. Think of all the record files as a sequence sorted by date. This function moves back one step in this sequence.

`records-goto-next-record-file`
: Key *C-c C-f*. Move to the next record file. See `records-goto-prev-record-file` for more details.

`records-goto-prev-day`
: Key *C-c C-y*. Go to the record file of the previous day (yesterday). If a record file does not exist, one will be created.

`records-goto-next-day`
: Key *C-c C-t*. Go to the record file of the next day (tomorrow). If a record file does not exist, one will be created.

`records-goto-link`
: Key *C-c C-g*. Goto the link around point in the records file. A link in a records file must be enclosed within `<` and `>`. This link can be a records link or a web link such as an http, FTP, gopher, news message link. It can also be an html link like `<a href=http://www.yahoo.com>` or `<http://www.yahoo.com>`. A records link identifies a record (see Section 5.1 [What is a Record], page 4). This record identifier can be copied to any other record and this function used to jump back to the record associated with the link. For instance, if you copied this link `<../../99/04/042199#* Work>` into some other record and then typed *C-c C-g* within the angle brackets, you would be transported to the `Work` record of April 21st, 1999. When the link is a web link, the link is handed off to `browse-url-browser-function`. When the link is a news message link, `deja.com` is searched for the news message.

  Note that C-g can be typed only after sometime has elapsed after typing C-c or else the command gets aborted because of the behavior of C-g. This is either an emacs bug or this binding ought to be changed.

`records-goto-mouse-link`
: Key *Button2*. When mouse is clicked on a link, goto the link. See `records-goto-link` for more details. When mouse is clicked anywhere else, invoke the default mouse binding.

`records-goto-last-record`
: Key *C-c C-l*. Go back to the last record file visited. Identical record files are not put in the history consecutively.

`records-search-forward`
: Key *C-c C-s*. Search forward for regular expression in current subject across record files.

`records-search-backward`
: Key *C-c C-r*. Search backward for regular expression in current subject across record files. Note: this key binding belonged to records-rename-record before version 1.4.6.

5.4 What is the Records Index File?

A records index file allows the records-mode to keep track of your records by date. The records index file can be accessed from within any record by typing *C-c C-j* (`records-goto-index`). Typing the same key sequence or typing ⟨RET⟩ will return you back to the current record. In the records index file, you can find out the dates on which you have written something about a subject. You can also type ⟨RET⟩ on any subject and date and jump to the appropriate record. TODO: The index file currently doesn't provide much functionality. Many more things can be added here (such as deleting records directly from the index file, greping within a subject, etc).

5.5 Adding, Deleting and Renaming Records

Records are manipulated with the following commands.

`records-insert-record`

Key *C-c C-i*. Insert a new record for the current date. Interactively asks for the subject. Subject completion is available with the ⟨TAB⟩ key.

You can insert templates when a new record is inserted. Look at the online documentation of the `records-insert-template` function and the `records-template-alist` variable. From Rob Tillotson, the author of the template code:

> In essence, with templates, records becomes a sort of ad-hoc database of journals, bibliographic information, contact logs, and all kinds of other random stuff. There might be advantages to keeping this kind of stuff in an SQL server or something, but records is convenient and easily greppable and doesn't require me to screw around with building databases (if I need to add a field, I can just type it in)...

`records-delete-record`

Key *C-c C-d*. Delete the current record for the current date. With argument, removes the subject only, so the body is preserved. If this is the last record on a subject, the user is asked if the subject should be removed from the index file. If the deletion of this record makes the file empty, the records file is deleted.

`records-rename-record`

Key *C-c C-e*. Renames the subject of the current record for the current date. Note: the key binding for this command has been changed in version 1.4.6.

`records-move-record`

Key *C-c C-m*. Moves the current record to another date. It prompts for the new date.

5.6 Miscellaneous Commands

`records-popup-mode-menu`

Key *Button3*. When mouse is clicked on a link, popup a link-specific menu. When mouse is clicked anywhere else, invoke the default popup menu.

`records-outline-mode`

`records-goto-calendar`
: Key *C-c C-c*. Goto the emacs calendar with the cursor on the date in the current record file.

`records-mark-record`
: Key *C-M-h*. Put mark at the end of the current record and point at the beginning of the record subject. With non-null prefix argument, the point is placed at the beginning of the record body.

`records-link-as-kill`
: Key *C-c C-k*. Put the link of the record around the current point in the kill ring. The link can be retrieved by the `yank` command (*C-y*). This function can be used to copy a record link to some other record.

`records-underline-line`
: Key *C-c -*. Underline the current line to the length of the line. This function is not really specific to records. You can use it to underline any text.

`records-insert-record-region`
: Key *C-x n r* Insert the region in the current buffer into today's record. Prompts for subject.

`records-insert-record-buffer`
: Key *C-x n b* Insert the current buffer into today's record. Prompts for subject.

`records-insert-link`
: Key *C-x n l* Writes the file name of the current buffer, or the url of the current page, or the message-id of the current gnus buffer as a records link at the end of today's record and inserts a user-specified comment. This function can be used to bookmark files, urls or news messages in your records file. Currently, this function is not bound to any key. You should bind it to a global key. This function used to be called records-insert-file-link.

5.7 Records Administration

Administration of records involves initialization and ensuring data and index consistency. This job is best done with an external (Perl) program called '`recordsadmin`'. The '`recordsadmin`' program performs the following jobs:

1. Installs or reinstalls user defaults. This is done with `recordsadmin -i`. The '`~/.emacs-records`' file is created or updated.
2. Converts one date and directory format to another. This is done with `recordsadmin -c`. Not everybody likes the same date format or the same directory format. Many different formats, such as 2/4 year, mm/dd/yy and dd/mm/yy formats are supported.
3. Re-indexes all the records. This is sort of a records fsck that ensures that the elisp indexing code hasn't blown things. Simply running '`recordsadmin`' does the indexing. Run `recordsadmin -h` to find out the usage.

6 Advanced Use

This section describes the advanced records commands.

6.1 Organizing your TODOs

The records TODO allows you to keep a list of things to be done per record. You start by creating a TODO such as the following inside a record.

```
* Work
------
link: <../../99/04/042199#* Work>

"
TODO: // created on 042199
some cool things to do
rotten things to do
END_TODO
"
```

For your convenience, the function `records-create-todo` can be used to create a new TODO. Key *C-c / t*. This function adds a comment with the date when the TODO was created.

The piece of text within the double quotes is called a TODO. Note that the double quotes are not part of the TODO. This TODO is written on the 21th of April, 1999. Suppose today is the 22th of April, 1999 and you open up a new record file today with `records-goto-today`. Then the TODO will be moved to today from the previous day. If you didn't create a record file on 22th, but created one on the 23th, the TODO would be moved from the 21th to the 23th. In general, a TODO for a subject is moved to today from the last time (day) you wrote a TODO for the subject. You can explicitly move TODOs with the function `records-get-todo`. Key *C-c / g*.

The match for a todo happens as follows:

```
^records-todo-begin-move-regexp
The todo text
^records-todo-end-regexp.*\n
```

The default value of records-todo-begin-move-regexp is "TODO: ". The default value of records-todo-end-regexp is "END_TODO". The comment added by `records-create-todo` is just part of the TODO text and can be removed.You can change these regexps if you like. Use *C-h v* to examine the values of these variables. You can also have multiple TODOs within a record. If you want a todo to be copied instead of moved, then use CTODO, instead of TODO. Look at `records-todo-begin-copy-regexp`. NOTE: the value of `records-todo-begin-move-regexp`, `records-todo-begin-copy-regexp`, `records-todo-end-regexp` has changed after records version 1.3. In fact, these variable are no longer regexps. So if you have changed them, beware.

The variable `records-todo-today` determines whether `records-goto-today` will do TODO processing. If set to true (the default), the TODO processing is done. If set to nil, the TODO processing is not done. If set to any other value, the user is asked whether

TODOs should be processed. The variables `records-todo-next-day` and `records-todo-prev-day` are similar to `records-todo-today` but are invoked when traversing to the next or the previous day. Their defaults are nil.

When a TODO is moved from a source to a destination record and, as a result, the source record becomes empty, then the record is deleted if the variable `records-todo-delete-empty-record` is set to true (the default).

6.2 Securing Records With Encryption

Consider the `Work` record of April 21st, 1999 (see Section 5.1 [What is a Record], page 4). That is a record I do not want my advisor to see! In records-mode, records can be individually encrypted (and signed) using Philip Zimmerman's PGP (Pretty Good Privacy) and the PGP mailcrypt mode available with `xemacs`. Since the mailcrypt mode is not available with `emacs` by default, it is included in the records distribution under the directory '`mailcrypt`'. However, this directory will probably not be in your load-path (See Info file '`xemacs`', node '`Loading`'). So if you don't have these packages by default, copy the files in the '`mailcrypt`' directory into the top level records directory. The mailcrypt version is 3.5.5 which supports PGP version 2.6, PGP version 5.0 and the GNU gpg.

By default PGP version 2.6 is used for encryption and decryption. You can set the value of the variable mc-default-scheme to mc-scheme-pgp50 to enable PGP version 5.0.

If you don't have PGP software then you either don't care about encryption or need to get it so you can use it with records.

`records-encrypt-record`
: Key *C-c / e*. Encrypt the current record for the current user. By default this function encrypts the whole record. With a prefix argument, start the encryption from point to the end of record. The function interactively asks whether the record should be signed also.

`records-decrypt-record`
: Key *C-c / d*. Decrypt the current record. The function asks for your PGP passphrase (and caches the passphrase for a short time (See Info file '`mailcrypt`', node '`Passphrase Cache`').

6.3 Putting Records Together

The records-mode allows concatenating several records together. These records are put together and output in a separate emacs buffer. Concatenation allows combining your thoughts. For instance, you can write a weekly status report of your work by combining your records on the `Work` subject for each of the days in the last week. This assumes that you write what you do everyday, which is the very reason you are using this software!

The records-mode allows concatenation of records and record files. Both are concatenated in date order.

`records-concatenate-records`
: Key *C-c + c*. Concatenate the current record with the records on the same subject written in the last *num* days. Output these records in the records

output buffer. The name of this buffer is contained in the variable `records-output-buffer`. Without prefix argument, prompts for number of days. An empty string will output the current record only. A negative number will output all the past records on the subject! Normally, the records are output in most-recent first order. This function asks the user if the order should be reversed.

`records-concatenate-records-latex`
: Key *C-c + l*. Concatenate the current record with the records on the same subject written in the last *num* days. Output these records in latex format in the records latex output buffer. The name of this buffer is contained in the variable `records-latex-output-buffer`. Then run latex on the output. Without prefix argument, prompts for number of days. An empty string will output the current record only. A negative number will output all the past records on the subject! Normally, the records are output in most-recent first order. This function asks the user if the order should be reversed.

`records-concatenate-record-files`
: Key *C-c + f*. Concatenate all the records in the records files of the last *num* days. All the records of a subject are collected together. Output these records in the records output buffer (see above). Without prefix argument, the function prompts for the number of days. An empty string will output the records of the current file.

`records-concatenate-record-files-latex`
: Key *C-c + r*. Concatenate all the records in the records files of the last *num* days. All the records of a subject are collected together. Output these records in latex format in the records latex output buffer (see above). Without prefix argument, the function prompts for the number of days. An empty string will output the records of the current file. This function is not implemented currently.

The records-mode concatenation related variables are shown below.

`records-output-buffer`
: Contains the output of concatenating records. By default, "*RECORDS-OUTPUT*".

`records-subject-prefix-on-concat`
: Prefix prepended to each subject on records concatenation. By default, "— ".

`records-subject-suffix-on-concat`
: Suffix appended to each subject on records concatenation. By default, " —".

`records-date-prefix-on-concat`
: Prefix prepended to each date on records concatenation. By default, "* "

`records-date-suffix-on-concat`
: Suffix appended to each date on records concatenation. By default, "".

`records-select-buffer-on-concat`
: If non-nil, the records-output-buffer is selected after records are concatenated by the concatenation functions. If nil, the records-output-buffer is just displayed. By default, nil.

`records-erase-output-buffer`
: If non nil, the records-output-buffer is erased, every time the concatenation functions are invoked. If nil, the output is appended. By default, nil.

`records-latex-output-buffer`
: Contains the output of concatenating records in latex format. By default, "*RECORDS-LATEX-OUTPUT*".

`records-tex-directory`
: Location of latex template for records. By default, "RECORDS-DIRECTORY/tex/".

`records-tex-template-plain`
: Location of plain template for LaTeX'ing records. By default, "RECORDS-DIRECTORY/tex/records-templ.tex".

`records-tex-temp-output-file`
: Output location for Latex runs. By default, "RECORDS-DIRECTORY/tex/records-temp.tex".

6.4 Editing Records In LaTeX mode

Records can be edited in LaTeX mode for those who insist on writing everything in this arcane language! The latex mode can be invoked in each record by calling `records-narrow-latex`. The record changes all links and TODOs to latex compatible environments. To get out of latex mode, use `records-widen-latex`.

`records-narrow-latex`
: Key *C-c / l*. Narrow records-file buffer to the current record and switch to latex-mode.

`records-widen-latex`
: Widen records-file buffer and switch from latex to records-mode. This key is not bound by default since the buffer is now in LaTeX mode. You will have to invoke this function in the minibuffer (`M-x records-widen-latex`) to come back to records mode. Or else you can bind the key *C-c l w* to this function in LaTeX mode as shown in the installation section (see Chapter 3 [Installation], page 1).

6.5 Editing Records In Outline mode

Records can be edited in Outline mode. The outline minor mode is toggled by calling `records-ouline-mode`.

`records-ouline-mode`
: Key *C-c / o*.

7 Conclusions and Acknowledgments

As promised, the *records-mode* for emacs allows creation of an online diary with extensive linking and indexing facilities. This diary has in fact become my major desktop tool. I write most of my documents, small and large, as records distributed over different days and records-mode does the tracking for me. If you have noticed, records provides an editor and a browser all in one. You can browse your records with automatically generated records links, and you can also edit the contents of your records.

This software was originally inspired by John Heidemann's notes-mode `http://www.isi.edu/~johnh/SOFTWARE/NOTES_MODE/index.html`. This version enhances the original notes system by adding several features that John hasn't had time to add. The main addition is that indexing is done on the fly so that indexes are current as you add or delete new records. John's notes mode updates indexes in the background (say, daily) and I found that inconvenient.

Kaarthik Sivakumar `sivakuma@mail.eecis.udel.edu` insisted on documentation and that forced me to write the info pages.

Please send me mail at `ashvin@users.sourceforge.net` if you find this software useful.

Variable Index

Concept and Function Index

Key Index

Table of Contents