

refdb-mode

Markus Hoenicka

refdb-mode

Markus Hoenicka

Abstract

An Emacs minor mode for the RefDB reference manager and bibliography tool.

Table of Contents

Preface	vii
1. Installation	8
1.1. Prerequisites	8
1.2. Installing the Emacs Lisp files	8
1.3. Setting up reftdb-mode	9
1.4. Customizing reftdb-mode	9
1.5. Invoking reftdb-mode	9
1.6. Using reftdb-mode with NTEmacs	10
2. Concepts	13
2.1. Default database	13
2.2. Data formats	13
2.2.1. RIS	13
2.2.2. risx	13
2.2.3. xnote	13
2.2.4. citestylex	14
2.2.5. Other formats	14
2.2.6. Selecting output formats	14
2.3. Reference data and extended note identification	14
2.4. Screen layout	15
2.5. Key bindings	15
2.6. reftdb-mode is schizophrenic	16
2.7. Working with reference data	17
2.7.1. Adding and updating references	17
2.7.2. Retrieving references	17
2.7.3. Converting reference data	18
2.8. Working with extended notes	18
2.9. Working with markup documents	18
3. reftdb-mode commands	20
3.1. Managing references	20
3.1.1. Add References	20
3.1.2. Update References	20
3.1.3. Check References	20
3.1.4. Delete References	20
3.1.5. Get References submenu	21
3.1.6. Get References on Region submenu	23
3.1.7. Get References from Citation	24
3.1.8. Pick References	24
3.1.9. Dump References	25
3.1.10. Convert References submenu	25
3.2. Managing notes	26
3.2.1. Add Notes	26
3.2.2. Update Notes	26
3.2.3. Delete Notes	26
3.2.4. Get Notes submenu	27
3.2.5. Get Notes on Region submenu	30
3.2.6. Add Links	31
3.2.7. Delete Links	31
3.3. Insert commands	31
3.3.1. Insert submenu	31
3.4. Reference formats and databases	32
3.4.1. Customize data output submenu	32
3.4.2. Select Database	34
3.4.3. Show Database Info	34

3.5. Managing documents	34
3.5.1. Create Document	34
3.5.2. Cite References submenu	34
3.5.3. Transform Document	36
3.5.4. View Output	36
3.6. Database management	36
3.6.1. Create Databases	36
3.6.2. List Databases	36
3.6.3. Delete Databases	37
3.7. Bibliography styles management	37
3.7.1. Add Styles	37
3.7.2. List Styles	37
3.7.3. Get Style	37
3.7.4. Delete Styles	37
3.8. User management	38
3.8.1. Add Users	38
3.8.2. List Users	38
3.8.3. Delete Users	38
3.9. Journal title word management	38
3.9.1. Add Journal Title Words	38
3.9.2. List Journal Title Words	39
3.9.3. Delete Journal Title Words	39
3.10. Miscellaneous server tasks	39
3.10.1. Run Keyword Scan	39
3.10.2. View Server Information	39
3.11. Database backup management	39
3.11.1. Backup Reference Databases	39
3.11.2. Restore Reference Databases	39
3.12. refdb-mode customization	40
3.12.1. Initialize System Database	40
3.12.2. Customize RefDB Mode	40
3.12.3. Edit RefDB Config Files	40
3.13. RefDB server management	41
3.13.1. Start Application Server on localhost	41
3.13.2. Stop Application Server on localhost	41
3.13.3. Restart Application Server on localhost	41
3.13.4. Reload Application Server on localhost	41
Index	43

List of Tables

2.1. reldb-mode key bindings	15
------------------------------------	----

Preface

refdb-mode turns GNU Emacs into a frontend for RefDB [<http://refdb.sourceforge.net>], a reference manager and bibliography tool for markup languages. refdb-mode is a minor mode which works well alongside the major modes PSGML (SGML editing), nxml-mode (XML editing), auctex (LaTeX editing), muse-mode (Muse and Planner), and ris-mode (RIS editing). It offers a full set of menu commands for reference and notes management, allowing you to conveniently use Emacs to create or update references and notes. Moreover, it allows you to create DocBook and TEI documents for use with RefDB. You can transform the documents and view the results with a mouseclick. It integrates seamlessly into your authoring process, allowing you to search for references and yank them directly in place, or to do reverse lookups from cited references, without ever leaving Emacs. Finally, there is also a full set of administrative commands to control all aspects of RefDB configuration and administration.

This manual gives a brief overview of how to use refdb-mode, followed by a more thorough description of all available commands.

Please visit the RefDB web page [<http://refdb.sourceforge.net>] for further information about RefDB and for the addresses of mailing lists where problems with refdb-mode can be discussed.

Chapter 1. Installation

Installing `refdb-mode` is not much different from installing other Emacs extensions. It is assumed that you have a basic understanding of the Emacs configuration and directory layout. For further information about this topic, please consult the GNU Emacs manual [<http://www.gnu.org/software/emacs/manual/>].

1.1. Prerequisites

First of all you need GNU Emacs itself. Emacs is available prepackaged for most contemporary operating systems, or it can be obtained as a source tarball from the GNU site [<http://directory.fsf.org/emacs.html>]. You also need a working installation of RefDB [<http://refdb.sourceforge.net>]. You need at least the clients on the same computer that will run `refdb-mode`. If you access the application server **refdbd** on a remote server, you can build the required tools on your computer by configuring RefDB with the `--disable-server` switch.

The RefDB clients must be fully operational on your computer. If you can't run the RefDB clients manually, `refdb-mode` won't be able to do it either. This means in particular that your client configuration files (`~/.refdbarc`, `~/.refdbcrc`, and `~/.refdbibrc`) must be set up properly, including a valid username/password combo. `refdb-mode` cannot ask for the password interactively (aside from the fact that this would be annoying), so this must be set in your configuration file. Please refer to the RefDB manual [<http://refdb.sourceforge.net/doc.html#manual>] for further instructions

The bibliography data conversion relies on the external `bibutils` tools [<http://www.scripps.edu/~cdputnam/software/bibutils/bibutils.html>].

1.2. Installing the Emacs Lisp files

The most convenient way to install the files on your system is to use the autotools-based installation. To this end, run the following commands in the directory which contains the `refdb-mode` sources:

```
root# ./configure
```

You should check the output of this command carefully. You'll get a warning if the required RefDB tools are not found, and if **sudo** is missing. You'll be able to proceed with the installation anyway, but in the first case you won't be able to do anything, in the second case you won't be able to use a part of the administrative interface. The script also tries to find applications that you can use as external viewers for PDF, Postscript, and RTF files. If the script can't find an application which you know is installed, you may have to include this app into your path before running **configure**, e.g. like this:

```
root# export PATH=$PATH:/usr/local/OpenOffice.org1.1.4/program
root# ./configure
```

The applications are used as system-wide defaults for the document viewers. However, each user can individually customize `refdb-mode` after the installation if she wants to use different applications for this purpose.

Once this is done, you can byte-compile the Emacs Lisp files and install all required files:

```
root# make
```



```
root# make install
```

The Lisp files proper will be installed into an appropriate `site-lisp` directory. The configuration file `refdb-mode-config.el` will be installed in your system configuration directory, which is either `/etc` or `/usr/local/etc` on most systems. The HTML and PDF versions of the documentation will be installed in your documentation directory, usually `/usr/share/doc/refdb-elisp` or `/usr/local/share/doc/refdb-elisp`. An info version of the manual is also installed. You can view it from the command line using `info refdb-mode` or from within Emacs with the RefDB → Show Manual menu command.

If you run more than one Emacs version, or if you want to test-drive `refdb-mode` without installing it system-wide, you can pass an alternative `site-lisp` directory to `configure` with the `--with-lispdir` option. Make sure that this directory is in your load path, e.g. by adding the following line to your `.emacs`:

```
(add-to-list 'load-path "c:/path/to/elisp/")
```

1.3. Setting up refdb-mode

In order to use `refdb-mode` and the associated `ris-mode` you'll have to load these modes either from your `.emacs` or from a global startup file (`default.el` or `site-start.el`). To this end, load the configuration file from one of the startup files like this:

```
(load-file "/usr/local/etc/refdb-mode-config.el")
```

Alternatively, you can copy the code in that file directly into one of your startup files. Please see the comments in the example configuration file for further instructions.

Note

Some package management systems use a `site-start.d` directory which holds this kind of Emacs startup files. You can put an appropriately named symlink to `refdb-mode-config.el` into this directory.

1.4. Customizing refdb-mode

`refdb-mode` can be configured entirely through the customization interface of Emacs. Either run the menu command RefDB → Administration → Customize RefDB mode or type `M-x customize` and browse to the RefDB section in the Applications section.

If you used the autotools-based package to install `refdb-mode`, the file `/usr/local/etc/refdb-mode-config.el` contains a few system-wide customizations which adapt `refdb-mode` to your local system. `./configure` checks for file viewers available on your system, and it also sets the directory containing the global RefDB configuration files. This file is the best place for system-wide `refdb-mode` customizations.

1.5. Invoking refdb-mode

If you use the configuration file mentioned above, `refdb-mode` will be automatically invoked whenever

you load a file type associated with the mode, e.g. an XML file or a RIS file. To load the mode manually, use either the menu command Tools → RefDB Mode or run **M-x refdb-mode**.

1.6. Using refdb-mode with NTEmacs

Windows is an extra challenge for refdb-mode, as we have to call Cygwin programs (which use Unix-style paths) from a Win32 application (which uses Win32-style paths) by running a Cygwin shell (which needs a different environment than the one it inherits from NTEmacs) through Lisp commands, leaving enough room for things to go wrong. It boils down to getting your `.emacs` right, and we'll show here step by step what it takes to see the light. You'll notice that the setup uses UTF-8 quite heavily, assuming that you've set up your RefDB databases accordingly. I've settled for this encoding as it gave me the simplest consistent setup possible. You may well experiment with other character encodings, but I hardly see any benefit in doing so.

Note

Some of the administrative functions currently do not work under NTEmacs. Affected are all functions which run **sudo** for authentication or which start an xterm to obtain database passwords.

Note

Path names need a little extra care on Windows. Save yourself a great deal of pain by adding all Win32 executables that you want to call (mainly viewers for RTF, PDF, and Postscript files) to your Windows PATH. In your `/usr/local/etc/refdb-mode-config.el` file, use the command names instead of the full paths of the viewing programs. Viewing HTML files is handled by `browse-url` which should be set up to use an external browser like Firefox. The Emacs w3 browser does not seem to grok the paths assembled by refdb-mode.

```
;; use Unix-style line endings
(setq-default buffer-file-coding-system 'undecided-unix)

;; xml files as well as files with "utf8" somewhere
;; in the path are opened as Unicode
(setq file-coding-system-alist
      (append (list
                (cons "\\*.xml$" 'utf-8)
                (cons "utf8" 'utf-8))
              file-coding-system-alist))
```

First we ask NTEmacs to use Unix-style line endings. The second expression makes sure that all XML files as well as all files with the string "utf8" in their name (e.g. `refdata-utf8.ris`) are loaded as Unicode files using the UTF-8 character encoding.

```
;; use bash as the default shell
(setq exec-path (cons "C:/Programme/cygwin/bin" exec-path))
(setq shell-file-name "bash")
(setenv "SHELL" shell-file-name)
(setenv "PATH" (concat (getenv "PATH") ";C:\\Programme\\cygwin\\bin" ";C:\\Programme\\cygwin\\bin"))
(setq explicit-shell-file-name shell-file-name)
(setq explicit-shell-args ('("--login" "-i")))
(setq shell-command-switch "-ic")
(setq w32-quote-process-args t)
(defun bash ()
  (interactive)
  (let ((binary-process-input t)
```

```
(binary-process-output nil))
(shell)))

(setq process-coding-system-alist (cons '("bash" . (utf-8-dos . utf-8-unix))
process-coding-system-alist))
```

This block of commands uses the Cygwin version of **bash** as the default shell. It also makes sure that a few additional directories are added to the path. Finally it tells NTEmacs that the data returned by the shell use the UTF-8 encoding as well. The paths shown here need to be adapted to your local installation.

Note

If you want to process SGML files, you'll have to set `SGML_CATALOG_FILES` once in your `.bashrc` using Unix-style paths (bash, when called from NTEmacs, doesn't seem to evaluate `/etc/profile`) to make the catalogs available to OpenJade et al., and once in your system (the usual clicking odyssey in the control panel) using the Win32 paths. PSGML will use the latter when running under NTEmacs.

```
;; refdb-mode and ris mode
(load-file "c:/Programme/emacsen/site-lisp/refdb-mode-config.el")
(setq refdb-default-ris-encoding 'utf-8-unix)
```

Now we load refdb-mode and ris-mode through the config file. Again, use the path of your local system here. The second command makes sure we use UTF-8 again to display results in RIS format.

```
;; nxml mode
(load "c:/Programme/emacsen/site-lisp/nxml-mode-20041004/rng-auto.el")
(setq auto-mode-alist
  (cons '("\\.\\(xml\\|xsl\\|rng\\|xhtml\\)\\\\" . nxml-mode)
    auto-mode-alist))
```

Finally, we also load nxml-mode to edit XML files. The setup of PSGML to edit SGML-files has been described elsewhere [<http://www.mhoenicka.de/software/sgml/sgmlnt.html>] (slightly outdated, though).

The final step to make nxml-mode work properly is to make the RefDB RelaxNG schemas known to the system. To this end, create or edit the file `schemas.xml` in your home directory. The contents for the RefDB schemas looks like this:

```
<locatingRules xmlns="http://thaiopensource.com/ns/locating-rules/1.0">
  <documentElement prefix="" localName="ris" typeId="RISX"/>
  <documentElement prefix="" localName="STYLESET" typeId="CiteStyleX"/>
  <documentElement prefix="" localName="xnoteset" typeId="XNote"/>

  <typeId id="RISX" uri="/Programme/cygwin/usr/local/share/refdb/dtd/risx.rnc"/>
  <typeId id="CiteStyleX" uri="/Programme/cygwin/usr/local/share/refdb/dtd/citesty
  <typeId id="XNote" uri="/Programme/cygwin/usr/local/share/refdb/dtd/xnote.rnc"/>
</locatingRules>
```

You may have to run the command `M-x customize-variableRETrng-schema-locating-filesRET` and add

the path of your `schemas.xml` to the existing list of schema files. *Do not* use a drive letter but start the path with a slash, like in the file itself.

Chapter 2. Concepts

We'll start by introducing a few concepts which should make it easier to understand how refdb-mode works. In general it is useful to have a basic understanding of RefDB and of its command-line clients. See the RefDB manual [<http://refdb.sourceforge.net/doc.html#manual>] for further information.

2.1. Default database

RefDB needs to know which reference database you want to work with (there might be more than one on your system). To this end, refdb-mode remembers the name of the database once it is set and uses it for all subsequent commands that access a reference database. You have two options to permanently set the default database:

- Add the following command to your `~/.emacs`: (`setq refdb-database "EXAMPLE"`). Replace "EXAMPLE" with the name of your reference database.

Use the Customization buffer to set the RefDB database.

If you want to switch databases during a session, use the RefDB → Select Database menu command. This command will pop up a list of available databases that you can choose from.

2.2. Data formats

RefDB accepts and creates a variety of data formats. Emacs has editing modes for most of these. We'll briefly discuss what to do with these types.

2.2.1. RIS

The RIS format is sort of an interchange format for bibliographic data that most commercial bibliography software can read and create. RefDB accepts this format as a native input format. It is a tagged format, i.e. each logical line starts with a tag which describes the contents of that line. refdb-mode ships with an additional mode called ris-mode (`ris.el`) which adds syntax highlighting as well as some basic editing support for this data format to Emacs. The RIS format is fairly compact and easy to understand. This format is a good choice as the default output format, as it is both convenient to read and easy to update the database with changes.

2.2.2. risx

This is an XML application which describes the RIS standard (with minor extensions) using a syntax familiar to anyone who writes XML or SGML documents. It is more verbose than the RIS format, but it can also be used to update the database directly if you want to commit changes. risx has an edge over RIS in terms of validation. If you have nxml-mode associated with XML files, refdb-mode will use this mode to display query results in RISX format as well as source files containing bibliographic data. nxml-mode contains a validating parser, so you have an immediate feedback whether or not your changes to the data are valid. refdb-mode will refuse to add or update XML data if nxml-mode indicates they are not valid. This format is a good choice if you are familiar with XML data anyway and if you are not put off by its verbosity.

2.2.3. xnote

This is another XML application which describes extended notes. This is the only input format available

for extended notes. These notes are not part of your bibliographic data, but they can be attached to a variety of objects in your reference database, like (obviously) bibliographic records, but also author names, periodicals, and keywords. Use this format to add and update extended notes. When using nxml-mode for XML data, you'll also get validation for free.

2.2.4. citestylex

This is yet another XML application which describes bibliography and citation styles. This is the only available import and export format for style data. When using nxml-mode for XML data, you'll also enjoy the benefits of real-time validation.

2.2.5. Other formats

All other formats can only be used to export or view reference and notes data. They are of limited use in the interactive use of refdb-mode, but come in handy if you need to export data for other purposes.

2.2.6. Selecting output formats

refdb-mode uses a few menu commands to select and customize the output format of reference and note queries (remember that there is no choice for extended notes and citation styles).

The RefDB → Customize Data Output → Select Reference Output Type menu command pops up a list of available output formats. After selecting one of the available formats, all future calls to a command which returns reference data will use this output format.

The RefDB → Customize Data Output → Select Notes Output Type menu command works exactly the same and selects the output format of extended notes.

The RefDB → Customize Data Output → Select Output Format menu command selects how much information is displayed in the reference data. RIS and risx always return all available data, as these formats are meant to add and update datasets in the database. The remaining formats show only the core bibliographic data. By default, they do not show information like the reprint status or the abstract. If you select All fields here, these formats will also display all information. If you select More (additional) fields, a customizable subset of the additional fields will be displayed. Use the RefDB → Customize Data Output → Select Additional Data Fields menu command to select the additional fields that you would like to be displayed. The IDs only can be used in conjunction with RIS output. Some commands of the command-line clients use this kind of minimized RIS data as input (this is currently not supported by refdb-mode).

2.3. Reference data and extended note identification

Each reference and note in a database carries two unique identifiers:

ID

A numeric identifier which is generated by RefDB automatically when adding a dataset. There is no way to alter the ID. An example of an ID is "513".

Citation key

An alphanumeric identifier which is generated by RefDB when adding a dataset unless the dataset specifies a citation key. If RefDB has to generate the citation key for a reference, it will use the last name of the first author and the publication year, followed by a character sequence to make the result database-unique if required. The result is something like "Miller1999b". Citation keys for exten-

ded notes are generated in a similar fashion from the note author's name and the timestamp.

2.4. Screen layout

Whenever you run `refdb-mode` commands which create output (as opposed to status messages which are displayed in the echo area), two windows open and display the results. The `*refdb-output*` buffer displays the command results proper, e.g. the bibliographic data which the commands in the Get References submenu create. `refdb-mode` tries to infer the proper major mode for the type of data, e.g. it will load `ris-mode` for RIS data and `nxml-mode` for any of the XML formats. The modeline shows the major mode used as well as the character encoding. The `*refdb-messages*` buffer keeps a log of the result summaries, e.g. the number of retrieved references.

Tip

If you want to preserve search results displayed in the `*refdb-output*` buffer for later use, simply rename the buffer: **M-x rename-buffer newname**. This temporary buffer is not associated with a file, although you can of course write it to a file using **C-x C-w**. Your next query will generate another buffer called `*refdb-output*`. You can repeat this process any number of times to preserve an unlimited number of query results.

Commands that require additional data will prompt for these in the minibuffer. You can use the minibuffer history to recover any previously used input. In most cases you can also use minibuffer completion (**tab** or **space** key) to fill in the remainder of an argument or to choose between alternatives.

2.5. Key bindings

The most common `refdb-mode` commands are accessible through keyboard shortcuts. The common prefix of these commands is **Ctrl-c Ctrl-r**. The remainder of the key sequences are grouped by the purpose of the commands in a similar fashion as in the menu. Reference and notes management functions use **r** and **n** as a prefix, respectively. Commands that retrieve references by exact match and by regexp match use **g** and **x** as a prefix, respectively. The corresponding prefixes for notes management are **o** and **p**. Retrieving references and notes on region requires the **Ctrl-g** and **Ctrl-o** prefixes. Insert commands use the **i** prefix, select commands use the **s** prefix, and commands related to transforming documents and citing references use **c** as a prefix.

Table 2.1. refdb-mode key bindings

common prefix	action	target
Ctrl-c Ctrl-r	modify references: r modify notes: n	add: a update: u check: c delete: d
Ctrl-c Ctrl-r	get reference (exact): g get reference (regexp): x get reference on region: Ctrl-g get note (exact): o	author: a title: t keyword: k periodical: p

common prefix	action	target
	get note (regexp): p get note on region: Ctrl-o	ID: i citation key: c advanced: d keywordlink: l IDlink: q citation key link: v
Ctrl-c Ctrl-r	insert: i	author: a keyword: k periodical: p
Ctrl-c Ctrl-r	select: s	database: d reference output type: r notes output type: n
Ctrl-c Ctrl-r	transform: c	transform document: c view document: v cite as DocBook from point: d cite as TEI from point: t cite as LaTeX from point: l cite as Muse from point: m cite as DocBook (region): s cite as TEI (region): t cite as LaTeX (region): x cite as Muse (region): y

2.6. refdb-mode is schizophrenic

refdb-mode uses two different accounts for the regular reference management commands on the one hand and for the administrative commands on the other hand. The latter are all in the Administration submenu. refdb-mode will automatically use the appropriate username and password if your client configuration files for **refdbc** (regular commands) and **refdba** (administrative commands) are set up properly.

Some administrative commands require root permissions. To this end, refdb-mode will run **sudo**. You or your site administrator will have to set up an appropriate **sudoers** file which will allow you to run commands as root, see sudo(8). refdb-mode will ask you for your sudoers password whenever it is re-

quired.

2.7. Working with reference data

Getting bibliographic references into and out of the database is a key feature of any bibliographic software. Sure enough `refdb-mode` has commands to add, update, and retrieve bibliographic data. In addition there are a few commands to convert bibliographic data from and to formats that RefDB does not support natively.

2.7.1. Adding and updating references

To add data from an external source, either load the source file or yank the data into an appropriate empty buffer. If set up correctly, Emacs will use an appropriate major mode to display and edit the data. You can now go ahead and make any changes to the data. `refdb-mode` commands work on the buffer data, not on the source file. That is, you can edit to your heart's content without having to change the original source files.

When you are ready to commit the data, you'll have to watch out for the following:

- `refdb-mode` will accept `risx` data only if `nxml-mode` declares the buffer valid (if you use a different mode to edit XML data, no check is available). You cannot add a subset of a buffer as RefDB requires you to send a complete, valid XML document.
- When working with RIS data, the commands to add or update references either work on a region or on the whole buffer if no region is selected. It is highly recommended to use `transient-mark-mode` with RIS data. If there is a region, it will be highlighted. To deactivate the mark and use the whole buffer instead, press **C-g**.

To add the contents of the buffer or of the selected region to the database, use the Add References menu command. If you have retrieved the data from the database and want to commit your changes, use the Update References menu command instead.

2.7.2. Retrieving references

`refdb-mode` provides a convenient interface to run simple queries. Simple queries are queries which select references by a single field only, e.g. by author or by keyword. In addition there is a command to type in complex queries which uses the same syntax as the `getref` command of RefDB.

You have to tell `refdb-mode` where to search for these values. The command names indicate the field to search in. Simple queries come in three flavors:

- You are prompted to type in the value to search for, which has to match exactly. These commands are in the Get References submenu. Minibuffer completion and history are available to enter or reuse query strings.
- You are prompted to type in the value, which is a regular expression. These commands are also in the Get References submenu, and completion as well as history are also available. Please be aware that you may have to escape some characters in order to generate valid query strings. The type of regular expression (Unix vs. SQL) depends on the database engine used by RefDB.
- You select a region with the keyboard or with your mouse and use that as a regular expression. These commands are in the Get References on Region submenu. This type of query comes in handy to look up words or phrases that you write in your document and see whether there are any references that contain these strings as keywords or in the title. It is also useful to display references that you have

cited somewhere in your text: just mark the citation key and run a query on this region.

If your query is successful, the `*refdb-output*` buffer will contain the matching references in the format that you've selected. The `*refdb-messages*` buffer will indicate the number of retrieved references. If the data are either in RIS or in risx format, you can edit the data right away and update the database.

2.7.3. Converting reference data

Reference data in several formats that RefDB does not understand natively can be converted to RIS. Load the data into a buffer and run the appropriate "From" command in the Convert References submenu.

Going the other way round, RIS data can be exported to formats that RefDB cannot readily create. Use the appropriate "To" command in the Convert References submenu to export the data to EndNote or to MODS.

2.8. Working with extended notes

Extended notes are XML documents that link a note text to one or more objects (references, keywords, author names, periodicals) in your reference database. You work with these data basically in the same way as with risx reference data. `nxml-mode` provides real-time validation, and the Add Notes and Update Notes commands also refuse to run if the buffer is not valid.

Retrieving notes also works in a similar fashion as retrieving references, but with an additional quirk: as notes are linked to database objects, you can use the link target as a criterion in notes queries. Wherever appropriate, minibuffer completion and history are also available. The relevant functions are located in the Get Notes and Get Notes on Region submenus.

2.9. Working with markup documents

RefDB implements a convenient way to edit and publish markup documents using bibliographies. The transformation and bibliography creation is controlled by a Makefile in the same directory as the source file. This is fully supported by `refdb-mode`.

If you want to create a document for use with RefDB from scratch, use the Create Document command. You'll be prompted for a few bits of information. Then RefDB will create a Makefile and appropriate skeleton documents in the desired format. The latter will be loaded into a buffer so you may edit it right away.

Note

By default, the document using the full citation format will be loaded as `refdb-mode` can handle the link normalization on its own. However, if you have a reason to do so you can configure `refdb-mode` to load the document using the short citation format.

To add citations to your document, use the appropriate Get References or Get References on Region commands to retrieve the references you want to cite. Then either move point into the reference that you want to cite, or set mark and point to select several references. Use the appropriate command of the Cite References submenu to add the citation(s) to the kill ring. Now switch back to your source document and use `C-y` to yank the citation in place. The default citation format works with XML documents. If you're editing SGML documents, use the RefDB → Customize Data Output → Select Citation Output Format menu command to switch to SGML.

It is also easy to do a reverse lookup, i.e. display the references in a citation. To this end, place point

somewhere into a citation and run the RefDB → Get References from Citation menu command.

Chapter 3. refdb-mode commands

Essentially all commands are accessible through the menu, and the most common commands are also accessible through keyboard shortcuts. The RefDB menu is subdivided into several sections. We'll discuss the sections in the order of appearance. Each description contains the synopsis of the refdb-mode command, i.e. what you'll have to type in after pressing **M-x** if you want to run the command from the minibuffer.

3.1. Managing references

This section contains all commands that you need to manage references.

3.1.1. Add References

```
refdb-addrref-on-region(void);
```

bound to: **Ctrl-c Ctrl-r ra**

Adds the references in the current buffer to the currently selected database. If the buffer is a risx buffer, the whole contents of the buffer will be added to the database. If nxml-mode considers the buffer to be invalid, an error message is returned and no data are sent to the database

If the buffer is a RIS buffer and no region is selected, the whole buffer will be sent to the database. If a region is selected, only the references contained in the region will be sent to the database.

3.1.2. Update References

```
refdb-updateref-on-region(void);
```

bound to: **Ctrl-c Ctrl-r ru**

Updates the references in the current buffer, i.e. existing references in the database are updated with the data of the current buffer. Works essentially like the Add References command.

3.1.3. Check References

```
refdb-checkref-on-region(void);
```

bound to: **Ctrl-c Ctrl-r rc**

Checks the references in the current buffer against the ones in the current database. You'll get a list of possible duplicate references, author names, periodicals, and keywords.

3.1.4. Delete References

```
refdb-deleteref(idlist);
```

bound to: **Ctrl-c Ctrl-r rd**

Asks interactively for a list of IDs to delete from the current database. The references will be permanently removed from the database.

Note

It is not possible to delete references that other users have added to their personal interest list. Use Dump References instead in this case

3.1.5. Get References submenu

This submenu provides a variety of commands to run simple and complex queries. The commands interactively ask for the value to search for. Wherever appropriate there is a choice between exact matches and regular expressions.

3.1.5.1. Author

```
refdb-getref-by-author(author);
```

bound to: **Ctrl-c Ctrl-r ga**

Asks interactively for the name of an author. All references that contain this person as an author, as an editor, or as a series editor will be displayed. refdb-mode attempts an exact match of the value that you type in. Tab completion is available.

3.1.5.2. Author (regexp)

```
refdb-getref-by-author-regexp(author);
```

bound to: **Ctrl-c Ctrl-r xa**

Asks interactively for the name of an author as a regular expression. All references that contain this person as an author, as an editor, or as a series editor will be displayed. refdb-mode attempts a regular expression match of the value that you type in. Tab completion is available.

3.1.5.3. Title

```
refdb-getref-by-title(title);
```

bound to: **Ctrl-c Ctrl-r gt**

Asks interactively for the title of a reference. All references that use this string as a title, a book title, or a series title will be displayed. refdb-mode attempts an exact match of the value that you type in.

3.1.5.4. Title (regexp)

```
refdb-getref-by-title-regexp(title);
```

bound to: **Ctrl-c Ctrl-r xt**

Asks interactively for a title as a regular expression. All references that contain a matching string as a title, a book title, or a series title will be displayed. refdb-mode attempts a regular expression match of the value that you type in.

3.1.5.5. Keyword

```
refdb-getref-by-keyword(keyword);
```

bound to: **Ctrl-c Ctrl-r gk**

Asks interactively for a keyword. All references that contain this keyword will be displayed. refdb-mode attempts an exact match of the value that you type in. Tab completion is available.

3.1.5.6. Keyword (regexp)

```
refdb-getref-by-keyword-regexp(keyword);
```

bound to: **Ctrl-c Ctrl-r xk**

Asks interactively for a keyword as a regular expression. All references that contain this keyword will be displayed. refdb-mode attempts a regular expression match of the value that you type in. Tab completion is available.

3.1.5.7. Periodical

```
refdb-getref-by-periodical(periodical);
```

bound to: **Ctrl-c Ctrl-r gp**

Asks interactively for the name of a periodical. All references that were published in this periodical will be displayed. refdb-mode attempts an exact match of the value that you type in. Tab completion is available.

3.1.5.8. Periodical (regexp)

```
refdb-getref-by-periodical-regexp(periodical);
```

bound to: **Ctrl-c Ctrl-r xp**

Asks interactively for the name of a periodical as a regular expression. All references that were published in this periodical will be displayed. refdb-mode attempts a regular expression match of the value that you type in. Tab completion is available.

3.1.5.9. ID

```
refdb-getref-by-id(id);
```

bound to: **Ctrl-c Ctrl-r gi**

Asks interactively for the numerical ID of a reference which will be displayed. The ID is a unique identifier of each reference which is maintained by RefDB.

3.1.5.10. Citation key

```
refdb-getref-by-citekey(citekey);
```

bound to: **Ctrl-c Ctrl-r gc**

Asks interactively for the alphanumeric citation key of a reference which will be displayed. The citation key is also a unique identifier which you can set when first adding a reference to the database.

3.1.5.11. Advanced search

```
refdb-getref-by-advanced-search(searchstring);
```

bound to: **Ctrl-c Ctrl-r gd**

Asks interactively for a complete query string. The syntax of **getref** queries is described in the RefDB manual [<http://refdb.sourceforge.net/manual/index.html>]. You may need to escape special characters to hide them from the Lisp interpreter in order to obtain a valid query string.

3.1.6. Get References on Region submenu

The commands of this submenu do not ask for a value to search for, but work on a region that you have selected in the current buffer, either by mouse or by setting mark and point appropriately. The contents of the region will be used as the value of your query.

3.1.6.1. Author

```
refdb-getref-by-author-on-region(void);
```

bound to: **Ctrl-c Ctrl-r Ctrl-g a**

Uses the selected region as the value of an author regular expression search. All references which contain the given person as an author, editor, or series editor will be displayed.

3.1.6.2. Title

```
refdb-getref-by-title-on-region(void);
```

bound to: **Ctrl-c Ctrl-r Ctrl-g t**

Uses the selected region as the value of a title regular expression search. All references which contain a matching string in the title, the book title, or the series title will be displayed.

3.1.6.3. Keyword

```
refdb-getref-by-keyword-on-region(void);
```

bound to: **Ctrl-c Ctrl-r Ctrl-g k**

Uses the selected region as the value of a keyword regular expression search. All references which contain a matching keyword will be displayed.

3.1.6.4. Periodical

```
refdb-getref-by-periodical-on-region(void);
```

bound to: **Ctrl-c Ctrl-r Ctrl-g p**

Uses the selected region as the value of a periodical regular expression search. All references which were published in a matching periodical will be displayed.

3.1.6.5. ID

```
refdb-getref-by-id-on-region(void);
```

bound to: **Ctrl-c Ctrl-r Ctrl-g i**

Uses the selected region as the value of an ID search. The reference with the given ID will be displayed.

3.1.6.6. Citation Key

```
refdb-getref-by-citekey-on-region(void);
```

bound to: **Ctrl-c Ctrl-r Ctrl-g c**

Uses the selected region as the value of a citation key search. The reference with the given citation key will be displayed.

3.1.7. Get References from Citation

```
refdb-getref-from-citation(void);
```

bound to: **Ctrl-c Ctrl-r Ctrl-c**

Uses the citation around point to retrieve all cited references in that citation from the database.

3.1.8. Pick References

```
refdb-pickref(idlist);
```

Asks interactively for a list of ID values. These references will be added to your personal reference list.

3.1.9. Dump References

```
refdb-dumpref(idlist);
```

Asks interactively for a list of ID values. These references will be removed from your personal reference list.

3.1.10. Convert References submenu

This submenu offers a variety of commands to convert reference data to and from RIS. All commands work on the data in the current buffer and do not access the RefDB database. Use these commands to import from or export to data formats that RefDB does not support natively. In the first case, the buffer must contain reference data in one of the formats BibTeX, copac, EndNote, ISI, Medline, or MODS. The result will be in RIS format. In the second case the buffer must contain RIS data. The output will be either EndNote or MODS.

3.1.10.1. From BibTeX

```
refdb-import-from-bibtex(void);
```

Converts BibTeX data to RIS data.

3.1.10.2. From copac

```
refdb-import-from-copac(void);
```

Converts copac data to RIS data.

3.1.10.3. From EndNote

```
refdb-import-from-endnote(void);
```

Converts EndNote data to RIS data.

3.1.10.4. From ISI

```
refdb-import-from-isi(void);
```

Converts ISI data to RIS data.

3.1.10.5. From MedLine

```
refdb-import-from-medline(void);
```

Converts Medline data to RIS data.

3.1.10.6. From MODS

```
refdb-import-from-mods(void);
```

Converts MODS data to RIS data.

3.1.10.7. To EndNote

```
refdb-export-to-endnote(void);
```

Converts RIS data to EndNote data.

3.1.10.8. To MODS

```
refdb-export-to-mods(void);
```

Converts RIS data to MODS data.

3.2. Managing notes

This section contains all commands that you need to manage extended notes.

3.2.1. Add Notes

```
refdb-addnote-on-buffer(void);
```

bound to: **Ctrl-c Ctrl-r na**

Adds the extended notes in the current buffer to the currently selected database. If nxml-mode considers the buffer to be invalid, an error message is returned and no data are sent to the database

3.2.2. Update Notes

```
refdb-updatenote-on-buffer(void);
```

bound to: **Ctrl-c Ctrl-r nu**

Updates the extended notes in the current buffer, i.e. existing notes in the database are updated with the data of the current buffer. Works essentially like the Add Notes command.

3.2.3. Delete Notes

```
refdb-deletenote(idlist);
```

bound to: **Ctrl-c Ctrl-r nd**

Asks interactively for a list of NIDs to delete from the current database. The notes will be permanently removed from the database.

Note

It is not possible to delete notes that you do not own.

3.2.4. Get Notes submenu

This submenu provides a variety of commands to run simple and complex queries to retrieve extended notes. The commands interactively ask for the value to search for. Wherever appropriate there is a choice between exact matches and regular expressions.

3.2.4.1. Title

```
refdb-getnote-by-title(title);
```

bound to: **Ctrl-c Ctrl-r ot**

Asks interactively for the title of a reference. All references that use this string as a title will be displayed. refdb-mode attempts an exact match of the value that you type in.

3.2.4.2. Title (regexp)

```
refdb-getnote-by-title-regexp(title);
```

bound to: **Ctrl-c Ctrl-r pt**

Asks interactively for a title as a regular expression. All notes that contain a matching string as a title will be displayed. refdb-mode attempts a regular expression match of the value that you type in.

3.2.4.3. Keyword

```
refdb-getnote-by-keyword(keyword);
```

bound to: **Ctrl-c Ctrl-r ok**

Asks interactively for a keyword. All extended notes that contain this keyword will be displayed. refdb-mode attempts an exact match of the value that you type in. Tab completion is available.

3.2.4.4. Keyword (regexp)

```
refdb-getnote-by-keyword-regexp(keyword);
```

bound to: **Ctrl-c Ctrl-r pk**

Asks interactively for a keyword as a regular expression. All extended notes that contain this keyword

will be displayed. refdb-mode attempts a regular expression match of the value that you type in. Tab completion is available.

3.2.4.5. Note ID

```
refdb-getnote-by-nid(id);
```

bound to: **Ctrl-c Ctrl-r oi**

Asks interactively for the numerical ID of an extended note which will be displayed. The ID is a unique identifier of each note which is maintained by RefDB.

3.2.4.6. Citation key

```
refdb-getnote-by-ncitekey(citekey);
```

bound to: **Ctrl-c Ctrl-r oc**

Asks interactively for the alphanumeric citation key of an extended note which will be displayed. The citation key is also a unique identifier which you can set when first adding a note to the database.

3.2.4.7. Linked to Author

```
refdb-getnote-by-authorlink(author);
```

bound to: **Ctrl-c Ctrl-r oa**

Asks interactively for the name of an author which will be used for an exact match. All notes which are linked to this author name will be displayed.

3.2.4.8. Linked to Author (regexp)

```
refdb-getnote-by-authorlink-regexp(author);
```

bound to: **Ctrl-c Ctrl-r pa**

Asks interactively for the name of an author which will be used for a regexp match. All notes which are linked to this author name will be displayed.

3.2.4.9. Linked to Periodical

```
refdb-getnote-by-periodicallink(periodical);
```

bound to: **Ctrl-c Ctrl-r op**

Asks interactively for the name of a periodical which will be used for an exact match. All notes which are linked to this periodical will be displayed.

3.2.4.10. Linked to Periodical (regexp)

```
refdb-getnote-by-periodicallink-regexp(periodical);
```

bound to: **Ctrl-c Ctrl-r pp**

Asks interactively for the name of a periodical which will be used for a regexp match. All notes which are linked to this periodical will be displayed.

3.2.4.11. Linked to Keyword

```
refdb-getnote-by-keywordlink(keyword);
```

bound to: **Ctrl-c Ctrl-r ol**

Asks interactively for a keyword which will be used for an exact match. All notes which are linked to this keyword will be displayed.

3.2.4.12. Linked to Keyword (regexp)

```
refdb-getnote-by-keywordlink-regexp(keyword);
```

bound to: **Ctrl-c Ctrl-r pl**

Asks interactively for a keyword which will be used for a regexp match. All notes which are linked to this keyword will be displayed.

3.2.4.13. Linked to ID

```
refdb-getnote-by-idlink(id);
```

bound to: **Ctrl-c Ctrl-r oq**

Asks interactively for the ID of a reference. All notes which are linked to this reference will be displayed.

3.2.4.14. Linked to Citation Key

```
refdb-getnote-by-citekeylink(citekey);
```

bound to: **Ctrl-c Ctrl-r ov**

Asks interactively for the citation key of a reference. All notes which are linked to this reference will be displayed.

3.2.4.15. Advanced Notes search

```
refdb-getnote-by-advanced-search(searchstring);
```

bound to: **Ctrl-c Ctrl-r od**

Asks interactively for a complete notes query string. The syntax of **getnote** queries is described in the RefDB manual [<http://refdb.sourceforge.net/manual/index.html>]. You may need to escape special characters to hide them from the Lisp interpreter in order to obtain a valid query string.

3.2.5. Get Notes on Region submenu

The commands of this submenu do not ask for a value to search for, but work on a region that you have selected in the current buffer, either by mouse or by setting mark and point appropriately. The contents of the region will be used as the value of your query.

3.2.5.1. Title

```
refdb-getnote-by-title-on-region(void);
```

bound to: **Ctrl-c Ctrl-r Ctrl-o t**

Uses the selected region as the value of a title regular expression search. All notes which contain the given title will be displayed.

3.2.5.2. Keyword

```
refdb-getnote-by-keyword-on-region(void);
```

bound to: **Ctrl-c Ctrl-r Ctrl-o k**

Uses the selected region as the value of a keyword regular expression search. All notes which contain a matching keyword will be displayed.

3.2.5.3. Linked to Author

```
refdb-getnote-by-authorlink-on-region(void);
```

bound to: **Ctrl-c Ctrl-r Ctrl-o a**

Uses the selected region as the value of an author name regular expression search. All notes which are linked to a matching person will be displayed.

3.2.5.4. Linked to Keyword

```
refdb-getnote-by-keywordlink-on-region(void);
```

bound to: **Ctrl-c Ctrl-r Ctrl-o l**

Uses the selected region as the value of a keyword regular expression search. All notes which are linked

to a matching keyword will be displayed.

3.2.5.5. Linked to Periodical

```
refdb-getnote-by-periodicallink-on-region(void);
```

bound to: **Ctrl-c Ctrl-r Ctrl-o p**

Uses the selected region as the value of a periodical search. Notes which are linked to a matching periodical will be displayed.

3.2.5.6. Linked to ID

```
refdb-getnote-by-idlink-on-region(void);
```

bound to: **Ctrl-c Ctrl-r Ctrl-o q**

Uses the selected region as the value of a reference ID search. Notes linked to the reference with the given ID will be displayed.

3.2.5.7. Linked to Citation Key

```
refdb-getnote-by-citekeylink-on-region(void);
```

bound to: **Ctrl-c Ctrl-r Ctrl-o v**

Uses the selected region as the value of a reference citation key search. Notes linked to the reference with the given citation key will be displayed.

3.2.6. Add Links

```
refdb-addlink(void);
```

Asks interactively for the ID of an extended note, and for a list of items to link to. Appropriate links will be added to the note.

3.2.7. Delete Links

```
refdb-deletelink(void);
```

Asks interactively for the ID of an extended note, and for a list of items to be unlinked. Matching links will be deleted from the note.

3.3. Insert commands

3.3.1. Insert submenu

This submenu contains commands to insert strings based on the completion lists for author names, keywords, and periodicals.

3.3.1.1. Insert Author

```
refdb-insert-author(author);
```

bound to: **Ctrl-c Ctrl-r ia**

Prompts for the name of an author. You can use minibuffer completion to select from the list of author names in the current database, or type in a new name.

3.3.1.2. Insert Keyword

```
refdb-insert-keyword(keyword);
```

bound to: **Ctrl-c Ctrl-r ik**

Prompts for a keyword. You can use minibuffer completion to select from the list of keywords in the current database, or type in a new keyword.

3.3.1.3. Insert Periodical

```
refdb-insert-periodical(periodical);
```

bound to: **Ctrl-c Ctrl-r ip**

Prompts for the name of a periodical. You can use minibuffer completion to select from the list of periodical names in the current database, or type in a new name.

3.4. Reference formats and databases

3.4.1. Customize data output submenu

This submenu offers commands to switch between different data output types for references and notes, as well as the output formats of reference data and of citations.

3.4.1.1. Select Reference Output Type

```
refdb-select-data-output-type(outputtype);
```

bound to: **Ctrl-c Ctrl-r sr**

Reference data can be retrieved in a variety of formats. If you use the menu command, a submenu with the available types will pop up. If you run the command in the minibuffer, please be aware that the argument is a symbol, not a string.

Note

RIS and RISX are the most convenient types for working with references, as they can be updated in the database without further conversion. The other types are useful for exporting the data.

3.4.1.2. Select Notes Output Type

```
refdb-select-notesdata-output-type(outputtype);
```

bound to: **Ctrl-c Ctrl-r sn**

Notes data can also be retrieved in several formats. If you use the menu command, a submenu with the available types will pop up. If you run the command in the minibuffer, please be aware that the argument is a symbol, not a string.

Note

xnote is the only type suitable for working with notes, as they can be updated in the database without further conversion. The other types are useful for exporting the data.

3.4.1.3. Select Output Format

```
refdb-select-data-output-format(format);
```

Selects the reference data output format. This will only affect output types other than RIS and RISX which always return all available information. With all other types only a reasonable subset is displayed by default. You can choose to display only ID values, all fields, or selected fields. If you run the command from the minibuffer, please note that format is a symbol, not a string.

3.4.1.4. Select Additional Data Fields

```
refdb-select-additional-data-fields(void);
```

Invokes the customization page for the appropriate variable which holds a list of data fields to display.

3.4.1.5. Select Citation Output Type

```
refdb-select-citation-type(citationtype);
```

Selects the output type of citations. The short and the full citation type are available. The default is full, and you should not change this unless you know what you're doing.

3.4.1.6. Select Citation Output Format

```
refdb-select-citation-format(citationformat);
```

Selects the output format of citations. The default is the XML syntax, but you can select the SGML syntax for SGML documents. The `xref` element is empty, therefore the XML version requires an additional slash to denote this fact.

3.4.2. Select Database

```
refdb-select-database(database);
```

bound to: **Ctrl-c Ctrl-r sd**

If used from the menu, this command will pop up a submenu with all databases that the current user has access to. The currently selected database is marked accordingly. Select one of the available databases to make it the active database.

If called interactively, this command will prompt the user with a completion list of available databases.

3.4.3. Show Database Info

```
refdb-whichdb(void);
```

Displays some information about the currently selected database.

3.5. Managing documents

This section of the menu contains the functions related to creating and publishing bibliographies.

3.5.1. Create Document

```
refdb-create-document(type);
```

This command will interactively ask for the information required to run the refdbnd script. When you run this command from the menu, it will pop up a submenu with the available document types. refdbnd will create a skeleton document and a Makefile which controls the transformation steps. Depending on the customization settings, refdb-mode will load the short or the full citation format document when done. You can edit this document right away.

3.5.2. Cite References submenu

This submenu contains the command to create citations of references and put them into the kill ring. The citations are then available to be yanked in your source document.

3.5.2.1. Current as DocBook

```
refdb-create-docbook-citation-from-point(void);
```

bound to: **Ctrl-c Ctrl-r cd**

Creates a DocBook citation element from the reference (in RIS or RISX format) containing point and puts it into the kill ring.

3.5.2.2. Current as TEI

`refdb-create-tei-citation-from-point(void);`

bound to: **Ctrl-c Ctrl-r ct**

Creates a TEI citation element from the reference (in RIS or RISX format) containing point and puts it into the kill ring.

3.5.2.3. Current as LaTeX

`refdb-create-latex-citation-from-point(void);`

bound to: **Ctrl-c Ctrl-r cl**

Creates a LaTeX citation element from the reference (in RIS or RISX format) containing point and puts it into the kill ring.

3.5.2.4. Current as Muse

`refdb-create-muse-citation-from-point(void);`

bound to: **Ctrl-c Ctrl-r cm**

Creates a Muse citation element from the reference (in RIS or RISX format) containing point and puts it into the kill ring.

3.5.2.5. In Region as DocBook

`refdb-create-docbook-citation-on-region(void);`

bound to: **Ctrl-c Ctrl-r cs**

Creates a DocBook citation element from the references (in RIS or RISX format) contained in the current region and puts it into the kill ring.

3.5.2.6. In Region as TEI

`refdb-create-tei-citation-on-region(void);`

bound to: **Ctrl-c Ctrl-r cr**

Creates a TEI citation element from the reference (in RIS or RISX format) contained in the current region and puts it into the kill ring.

3.5.2.7. In Region as LaTeX

`refdb-create-latex-citation-on-region(void);`

bound to: **Ctrl-c Ctrl-r cx**

Creates a LaTeX citation element from the reference (in RIS or RISX format) contained in the current region and puts it into the kill ring.

3.5.2.8. In Region as Muse

```
refdb-create-muse-citation-on-region(void);
```

bound to: **Ctrl-c Ctrl-r cy**

Creates a Muse citation element from the reference (in RIS or RISX format) contained in the current region and puts it into the kill ring.

3.5.3. Transform Document

```
refdb-transform(type);
```

bound to: **Ctrl-c Ctrl-r cc**

Transforms the current document to an output format. If run from the menu, a submenu with the available targets will pop up. This command will run the appropriate Makefile target to achieve the transformation. Before the actual transformation takes place, the linkends will be normalized and the user is asked to save any changes.

3.5.4. View Output

```
refdb-view-output(type);
```

bound to: **Ctrl-c Ctrl-r cv**

Runs an appropriate viewer for the output generated by Transform Document. If run from the menu, a submenu with the available output types will pop up. The viewer will usually run in a separate window.

3.6. Database management

Commands to manage reference databases.

3.6.1. Create Databases

```
refdb-createdb(void);
```

Asks interactively for the name of a new reference database and for the default character encoding. After the database is created, you may have to run Add Users to grant users access to the new database.

3.6.2. List Databases

```
refdb-listdb(dbname) ;
```

Asks interactively for a regular expression used to list reference databases. An empty string causes refdb-mode to list all available reference databases.

3.6.3. Delete Databases

```
refdb-deletedb(dbname) ;
```

Asks interactively for the name of an existing reference database that should be deleted. Minibuffer completion against the list of existing databases is available.

3.7. Bibliography styles management

This menu section contains all functions to work with bibliography style data.

3.7.1. Add Styles

```
refdb-addstyle-on-buffer(void) ;
```

Adds the XML data using the citestylex.dtd in the current buffer to the system database. refdb-mode will refuse to add the data if nxml-mode indicates they are invalid. If the database already contains a style with the given name, it will be replaced with the new version.

3.7.2. List Styles

```
refdb-liststyle(stylename) ;
```

Asks interactively for a regular expression used to list bibliography styles. An empty string causes refdb-mode to list all available styles.

3.7.3. Get Style

```
refdb-getstyle(stylename) ;
```

Asks interactively for the name of a bibliography style which is then retrieved in XML format. Minibuffer completion against the list of existing styles is available.

3.7.4. Delete Styles

```
refdb-deletestyle(stylename) ;
```

Asks interactively for the name of a bibliography style that should be deleted. Minibuffer completion against the list of existing styles is available.

3.8. User management

These functions provide an interface to the RefDB user management.

3.8.1. Add Users

```
refdb-adduser(void);
```

Interactively asks for the name of the database access should be granted to, the hostname from which connections should be allowed, the database username, and an optional database password if the user is not yet known to the database engine. The user will be registered as a RefDB user using these data. She will have access to the system database and to the given reference database.

Note

The hostname is required only for MySQL. It is the name of the computer where `refdbd`, the RefDB daemon runs. If it runs on your local computer, the proper value may be "localhost" or the true hostname/IP address, depending on the OS and the database engine.

3.8.2. List Users

```
refdb-listuser(void);
```

Asks interactively for the name of the database and for a regular expression used to list users which have references or notes in that database. Minibuffer completion is available against the list of available databases. An empty string as the username causes `refdb-mode` to list all available users of the given database.

3.8.3. Delete Users

```
refdb-deleteuser(void);
```

Asks interactively for the name of a user whose access to a reference database should be revoked. Additionally, the database name and the hostname must be specified. Minibuffer completion against the list of existing databases is available.

3.9. Journal title word management

Commands to maintain the word list used to identify unabbreviated words in journal names.

3.9.1. Add Journal Title Words

```
refdb-addword(wordlist);
```

Asks interactively for a whitespace-separated list of words which are unabbreviated journal title words. These words will be added to the system database.

3.9.2. List Journal Title Words

```
refdb-listword(word);
```

Asks interactively for a regular expression used to list unabbreviated journal title words. An empty string causes refdb-mode to list all available journal title words.

3.9.3. Delete Journal Title Words

```
refdb-deleteword(wordlist);
```

Asks interactively for a list of words that is to be deleted from the list of unabbreviated journal title words.

3.10. Miscellaneous server tasks

These are commands that don't really fit anywhere else.

3.10.1. Run Keyword Scan

```
refdb-scankw(void);
```

Initiates a background keyword scan.

3.10.2. View Server Information

```
refdb-viewstat(void);
```

Displays the RefDB server status information.

3.11. Database backup management

You should backup your precious reference and notes data once in a while. The commands in this menu section provide a simple interface for this task.

3.11.1. Backup Reference Databases

```
refdb-backup-database(void);
```

Asks interactively for the name of a reference database and creates an SQL dump of the data. Minibuffer completion is available for selecting the database. Asks for the name of the file to save the data to.

3.11.2. Restore Reference Databases

```
refdb-restore-database(void);
```

Asks interactively for the name of the database and for the name of a reference database SQL dump file. Minibuffer completion is available for selecting the file. A database with the given name will be created, and the data from the dump file will be filled in.

Warning

Before creating the database, an existing database with the same name will be dropped.

3.12. refdb-mode customization

These commands help you to set up RefDB in the first place and to adapt RefDB and refdb-mode to your needs.

3.12.1. Initialize System Database

```
refdb-init-refdb(void);
```

Initializes the system database. Use this command once to initialize your RefDB installation.

Warning

This command will destructively replace any existing RefDB system database.

3.12.2. Customize RefDB Mode

This menu command invokes the Emacs customization buffer for RefDB.

3.12.3. Edit RefDB Config Files

The commands in this submenu open one of the RefDB configuration files in Emacs to change the persistent configuration of the RefDB applications. User configuration files will be opened in a new window in the same Emacs frame. Global configuration files will be opened in another Emacs instance which is started using **sudo** to obtain the required permissions.

```
refdb-edit-refdbarc(void);
```

Edit the **refdba** user configuration file.

```
refdb-edit-refdbcrc(void);
```

Edit the **refdbc** user configuration file.

```
refdb-edit-refdbibrc(void);
```


Edit the **refdbib** user configuration file.

```
refdb-edit-global-refdbarc(void);
```

Edit the global **refdba** configuration file.

```
refdb-edit-global-refdbcrc(void);
```

Edit the global **refdbc** configuration file.

```
refdb-edit-global-refdbibrc(void);
```

Edit the global **refdbib** configuration file.

```
refdb-edit-global-refdbdrc(void);
```

Edit the global **refdbd** configuration file.

3.13. RefDB server management

These commands allow to control the RefDB application server. This works only if **refdbd** runs on your local computer. All of these commands require root permissions.

3.13.1. Start Application Server on localhost

```
refdb-start-server(void);
```

Start the **refdbd** application server.

3.13.2. Stop Application Server on localhost

```
refdb-stop-server(void);
```

Stop the **refdbd** application server.

3.13.3. Restart Application Server on localhost

```
refdb-restart-server(void);
```

Restart the **refdbd** application server. The running instance terminates, and a new instance is started.

3.13.4. Reload Application Server on localhost

refdb-reload-server(void);

Causes the running instance of the **refdbd** application server to read its configuration file again.

Index

A

refdb-addlink, 31
refdb-addnote-on-buffer, 26
refdb-addrf-by-title, 21
refdb-addrf-on-region, 20
refdb-addstyle-on-buffer, 37
refdb-adduser, 38
refdb-addword, 38

B

refdb-backup-database, 39

C

refdb-checkref-on-region, 20
refdb-create-docbook-citation-from-point, 34
refdb-create-docbook-citation-on-region, 35
refdb-create-document, 34
refdb-create-latex-citation-from-point, 35
refdb-create-latex-citation-on-region, 35
refdb-create-muse-citation-from-point, 35
refdb-create-muse-citation-on-region, 36
refdb-create-tei-citation-from-point, 34
refdb-create-tei-citation-on-region, 35
refdb-createdb, 36

D

refdb-deletedb, 37
refdb-deletelink, 31
refdb-deletenote, 26
refdb-deleteref, 20
refdb-deletestyle, 37
refdb-deleteuser, 38
refdb-deleteword, 39
refdb-dumpref, 25

E

refdb-edit-global-refdbarc, 40
refdb-edit-global-refdbarc, 40
refdb-edit-global-refdbdrc, 40
refdb-edit-global-refdbibrc, 40
refdb-edit-refdbarc, 40
refdb-edit-refdbarc, 40
refdb-edit-refdbibrc, 40
refdb-export-to-endnote, 26
refdb-export-to-mods, 26

G

refdb-getnote-by-advanced-search, 29
refdb-getnote-by-authorlink, 28
refdb-getnote-by-authorlink-on-region, 30
refdb-getnote-by-authorlink-regexp, 28
refdb-getnote-by-citekeylink, 29

refdb-getnote-by-citekeylink-on-region, 31
refdb-getnote-by-idlink, 29
refdb-getnote-by-idlink-on-region, 31
refdb-getnote-by-keyword, 27
refdb-getnote-by-keyword-on-region, 30
refdb-getnote-by-keyword-regexp, 27
refdb-getnote-by-keywordlink, 29
refdb-getnote-by-keywordlink-on-region, 30
refdb-getnote-by-keywordlink-regexp, 29
refdb-getnote-by-ncitekey, 28
refdb-getnote-by-nid, 28
refdb-getnote-by-periodicallink, 28
refdb-getnote-by-periodicallink-on-region, 31
refdb-getnote-by-periodicallink-regexp, 29
refdb-getnote-by-title, 27
refdb-getnote-by-title-on-region, 30
refdb-getnote-by-title-regexp, 27
refdb-getref-by-advanced-search, 23
refdb-getref-by-author, 21
refdb-getref-by-author-on-region, 23
refdb-getref-by-author-regexp, 21
refdb-getref-by-citekey, 23
refdb-getref-by-citekey-on-region, 24
refdb-getref-by-id, 22
refdb-getref-by-id-on-region, 24
refdb-getref-by-keyword, 22
refdb-getref-by-keyword-on-region, 23
refdb-getref-by-keyword-regexp, 22
refdb-getref-by-periodical, 22
refdb-getref-by-periodical-on-region, 24
refdb-getref-by-periodical-regexp, 22
refdb-getref-by-title-on-region, 23
refdb-getref-by-title-regexp, 21
refdb-getref-from-citation, 24
refdb-getstyle, 37

I

refdb-import-from-bibtex, 25
refdb-import-from-copac, 25
refdb-import-from-endnote, 25
refdb-import-from-isi, 25
refdb-import-from-medline, 25
refdb-import-from-mods, 26
refdb-init-refdb, 40
refdb-insert-author, 32
refdb-insert-keyword, 32
refdb-insert-periodical, 32

L

refdb-listdb, 36
refdb-liststyle, 37
refdb-listuser, 38
refdb-listword, 39

P

refdb-pickref, 24

R

refdb-reload-server, 41
refdb-restart-server, 41
refdb-restore-database, 39

S

refdb-scankw, 39
refdb-select-additional-data-fields, 33
refdb-select-citation-format, 33
refdb-select-citation-type, 33
refdb-select-data-output-format, 33
refdb-select-data-output-type, 32
refdb-select-database, 34
refdb-select-notesdata-output-type, 33
refdb-start-server, 41
refdb-stop-server, 41

T

refdb-transform, 36

U

refdb-updatenote-on-buffer, 26
refdb-updateref-on-region, 20

V

refdb-view-output, 36
refdb-viewstat, 39

W

refdb-whichdb, 34