

refdb handbook

covers version 0.9.6

Markus Hoenicka

`mhoenicka@users.sourceforge.net`

refdb handbook: covers version 0.9.6

by Markus Hoenicka

Revision History

Revision 1.20	2005-11-14
Revision 1.19	2004-12-12
Revision 1.18	2004-02-18
Revision 1.17	2003-12-20
Revision 1.16	2003-08-15
Revision 1.15	2003-02-16
Revision 1.14	2002-11-29
Revision 1.13	2002-11-11
Revision 1.12	2002-9-1
Revision 1.11	2002-7-23
Revision 1.10	2002-5-13
Revision 1.9	2002-3-2
Revision 1.8	2002-1-1
Revision 1.7	2001-11-22
Revision 1.6	2001-10-1
Revision 1.5	2001-8-9
Revision 1.4	2001-8-5
Revision 1.3	2001-7-21
Revision 1.2	2001-6-25
Revision 1.1	2001-6-18
Revision 1.0	2001-5-22

Table of Contents

Preface	xii
1. refdb in a nutshell.....	xii
2. Organization of this manual	xii
3. Further information	xii
I. Overview	xiv
1. Introduction	1
1.1. What is refdb?.....	1
1.2. Who should use refdb?	1
1.3. General features	2
1.4. Reference management features	3
1.5. Bibliography tool features	4
1.5.1. SGML and XML.....	4
1.5.2. LaTeX	5
1.6. Credits.....	5
2. System requirements	7
2.1. Operating system	7
2.2. Libraries necessary to build and run refdb	7
2.2.1. System libraries.....	7
2.2.2. Other libraries	8
2.3. Perl modules	8
2.4. External applications	9
3. Current limitations and platform issues	13
3.1. Limitations.....	13
3.2. Platform issues.....	13
II. Administrator manual.....	15
4. Installation.....	16
4.1. Quickstart guide.....	16
4.2. Upgrading from an older version.....	16
4.2.1. 0.9.5.....	16
4.2.2. 0.9.4.....	17
4.2.3. 0.9.3.....	17
4.2.4. 0.9.....	18
4.3. Things to know before you start	18
4.3.1. Which database server?.....	19
4.3.2. Where do the components go?.....	19
4.3.3. The mystery of the configuration files	22
4.3.4. Environment variables	24
4.3.5. Some notes on the filesystem.....	24
4.4. Installation on Linux and other Unix variants	25
4.4.1. Generic instructions	25
4.4.2. OS-specific hints	33
4.5. Installation on Windows NT/2000/XP	34

4.5.1. Prerequisites	35
4.5.2. Installation from the sources	36
4.5.3. Installation of prebuilt binaries	39
4.6. Installation on Windows 95/98/ME.....	40
4.7. Other operating systems	40
4.8. Finishing the refdb installation.....	40
4.8.1. Configuring your database server	40
4.8.2. Creating the databases.....	41
4.8.3. Configuration files.....	43
4.8.4. The SGML/XML support files.....	43
4.8.5. The shell scripts	43
4.9. Testing your installation	44
5. refdb administration	47
5.1. Create a database	47
5.2. Delete a database	47
5.3. Add or remove a user.....	47
5.3.1. Add users	48
5.3.2. Remove users	49
5.4. Configure the application server.....	50
5.4.1. Configuration at startup.....	50
5.4.2. Reconfiguring a running refdbd process	50
5.5. Add, edit, or delete bibliography styles.....	51
5.6. Logging data	52
5.6.1. What to log.....	52
5.6.2. Destinations.....	52
5.6.3. Log levels	53
5.6.4. Interpreting the log information.....	54
5.7. Security issues	55
5.7.1. Passwords.....	55
5.7.2. Database server access control.....	55
5.8. How to run several refdb instances.....	57
5.9. Backup your data	58
5.9.1. Creating a risx dump.....	59
5.9.2. Creating a SQL dump	59
III. User manual.....	61
6. Overview of the refdb clients	62
6.1. Quickstart guide.....	62
6.2. The interactive mode of refdba and refdbc.....	63
6.3. The non-interactive mode of refdba and refdbc	64
6.3.1. Example	65
6.4. The non-interactive mode of refdbib	65
6.5. Common command-line options for all clients	65
7. Data input.....	68
7.1. Writing RIS datasets.....	68
7.1.1. Overview	68

7.1.2. Character encodings.....	69
7.1.3. RIS tags.....	69
7.1.4. Examples.....	75
7.2. The Emacs RIS mode.....	77
7.3. Input filters.....	78
7.4. Writing risx datasets.....	79
7.5. Writing extended notes.....	80
7.6. Input data mangling.....	82
7.6.1. Information that refdb generates for you.....	82
7.6.2. Information that refdb mangles.....	83
8. Reference management.....	85
8.1. Add references.....	85
8.1.1. RIS datasets.....	85
8.1.2. risx datasets.....	86
8.2. Find and view references.....	86
8.3. Delete references.....	87
8.3.1. Example.....	87
8.4. Edit references.....	88
8.5. Print references.....	88
8.6. Use the personal reference list.....	88
8.7. Global edit references.....	89
8.8. Create periodical synonyms.....	90
8.9. Character encoding issues.....	90
8.9.1. Character encodings of databases.....	90
8.9.2. Character encodings of imported data.....	91
8.9.3. Character encodings of exported data.....	91
8.10. Use pdfroot.....	91
8.11. Interaction with external applications.....	92
8.11.1. Editor.....	92
8.11.2. Viewer.....	93
9. Notes management.....	94
9.1. Add extended notes.....	94
9.1.1. Example.....	94
9.2. Find and view extended notes.....	94
9.3. Delete extended notes.....	94
9.3.1. Example.....	95
9.4. Edit extended notes.....	95
9.5. Link existing notes to other objects in the database.....	95
9.5.1. Example.....	95
9.6. To share or not to share extended notes.....	95
9.6.1. The refdbd default.....	96
9.6.2. The <code>share</code> attribute of the extended notes.....	96
9.6.3. Data privacy.....	96
10. Bibliographies.....	98
10.1. Quickstart guide.....	98

10.2. Manage bibliography styles.....	98
10.2.1. Write or modify a bibliography style file.....	99
10.3. Create bibliographies.....	100
10.3.1. SGML and XML documents	100
10.3.2. Other SGML or XML document types.....	112
10.3.3. LaTeX/BibTeX.....	112
IV. Reference manual.....	115
11. refdbd: the application server.....	116
11.1. Overview	116
11.2. Starting refdbd	116
11.2.1. Using the control script.....	117
11.2.2. Start refdbd directly	117
11.3. The refdbdrc variables	121
12. refdba: the administrative client.....	125
12.1. Overview	125
12.2. Starting refdba	125
12.3. The refdbarc variables	125
12.4. Commands.....	127
12.4.1. addstyle	127
12.4.2. adduser	128
12.4.3. addword.....	130
12.4.4. confserv.....	131
12.4.5. createdb	132
12.4.6. deletedb.....	134
12.4.7. deletestyle	134
12.4.8. deleteuser	135
12.4.9. deleteword.....	136
12.4.10. getstyle	137
12.4.11. help.....	138
12.4.12. listdb.....	138
12.4.13. liststyle	139
12.4.14. listuser	139
12.4.15. listword	140
12.4.16. scanlw.....	141
12.4.17. set	141
12.4.18. verbose	142
12.4.19. viewstat	143
13. refdbc: the reference management client	144
13.1. Overview	144
13.2. Starting refdbc	144
13.3. The refdbc configuration file variables.....	145
13.4. Commands.....	148
13.4.1. addlink.....	148
13.4.2. addnote.....	149
13.4.3. addref	150

13.4.4. deletelink.....	151
13.4.5. deletenote.....	152
13.4.6. deleteref.....	153
13.4.7. dumpref.....	154
13.4.8. getau, geted, getas.....	155
13.4.9. getjo, getjf, getj1, getj2.....	156
13.4.10. getkw.....	157
13.4.11. getnote.....	158
13.4.12. getref.....	159
13.4.13. help, ?.....	161
13.4.14. listdb.....	161
13.4.15. liststyle.....	162
13.4.16. pickref.....	163
13.4.17. selectdb.....	163
13.4.18. set.....	164
13.4.19. updatenote.....	165
13.4.20. updateref.....	165
13.4.21. verbose.....	166
13.4.22. whichdb.....	167
13.5. Reference data output formats.....	168
13.5.1. scrn.....	168
13.5.2. html.....	168
13.5.3. xhtml.....	169
13.5.4. ris.....	169
13.5.5. risx.....	169
13.5.6. bibtex.....	169
13.5.7. db31.....	169
13.5.8. db31x.....	170
13.5.9. teix.....	170
13.6. Extended notes output formats.....	170
13.6.1. scrn.....	170
13.6.2. html.....	170
13.6.3. xhtml.....	170
13.6.4. xnote.....	170
13.7. The query language.....	171
13.7.1. The reference query language.....	171
13.7.2. The notes query language.....	175
13.7.3. Some example queries.....	176
13.8. Regular expressions.....	178
13.8.1. Unix-style regular expressions.....	178
13.8.2. SQL regular expressions.....	180
14. refdbib: the bibliography client.....	181
14.1. Overview.....	181
14.2. Starting refdbib.....	181
14.3. Examples.....	182

14.4. The refdbibrc variables	183
15. Reference data conversion tools	187
15.1. dos2unix	187
15.2. med2ris	187
15.2.1. Starting med2ris	187
15.2.2. The med2ris configuration variables	188
15.2.3. med2ris' behind-the-scenes data mangling	189
15.3. nmed2ris	190
15.3.1. Starting nmed2ris	190
15.3.2. The nmed2ris configuration variables	191
15.3.3. nmed2ris' behind-the-scenes data mangling	192
15.3.4. Data sources	193
15.4. bib2ris	193
15.4.1. Starting bib2ris	194
15.4.2. The bib2ris variables	196
15.4.3. bib2ris' data mangling	200
15.4.4. Post-processing with tex2mail	201
15.5. en2ris	202
15.5.1. Starting en2ris	202
15.5.2. The en2ris configuration variables	203
15.5.3. en2ris' behind-the-scenes data mangling	203
15.6. db2ris	204
15.6.1. Starting db2ris	204
15.6.2. The db2risrc variables	205
15.6.3. db2ris' data mangling	205
15.6.4. Modifying db2ris	206
15.7. marc2ris	206
15.7.1. Starting marc2ris	207
15.7.2. marc2ris data mangling	208
15.7.3. The marc2ris configuration variables	209
15.8. Convert SGML and XML data to risx	210
16. Bibliography tools	211
16.1. The refdbnd shell script	211
16.2. The runbib shell script	212
16.2.1. DocBook SGML output	213
16.2.2. DocBook and TEI XML output	213
16.2.3. BibTeX output	214
16.2.4. Example	214
16.3. The refdbjade shell script	214
16.3.1. Starting refdbjade	215
16.3.2. Resolution of public identifiers	215
16.3.3. Examples	216
16.4. The refdbxml shell script	216
16.4.1. Starting refdbxml	216
16.4.2. Configuring refdbxml	217

16.4.3. Resolution of public identifiers	218
16.4.4. Examples.....	218
16.5. The bibdb2tei.xml stylesheet	218
16.6. refdbxp.....	219
16.6.1. Starting refdbxp	220
16.6.2. Examples.....	220
V. Programmer's manual.....	222
17. Reference database design	223
17.1. The table t_meta	223
17.2. The table t_refdb.....	223
17.3. The table t_author.....	226
17.4. The table t_keyword	227
17.5. The table t_periodical	227
17.6. The table t_user	228
17.7. The table t_xauthor.....	228
17.8. The table t_xkeyword	229
17.9. The table t_xuser	229
18. The refdb database design	231
18.1. The table t_journal_words	231
18.2. The table CITSTYLE	231
18.3. The table REFSTYLE	231
18.4. The table SEPARATORS	231
18.5. The table POSITION	231
19. RIS and risx format specifications	233
19.1. The refdb RIS implementation	233
19.2. The risx DTD.....	233
20. The Perl client module	235
20.1. Installation	235
20.2. Classes and their functions	235
20.2.1. RefDBClient::Risdata	236
20.2.2. RefDBClient::Simplelist.....	236
20.2.3. RefDBClient::Enigma.....	237
20.2.4. RefDBClient::Client	237
21. Using the backend API to implement new output formats	253
21.1. The backend API	253
21.2. How to insert a new backend into refdb	253
22. The refdb SGML/XML input and output formats	254
22.1. The XML input format for bibliographies.....	254
22.2. The XML input format for bibliographic styles	254
22.3. The XML input format for extended notes.....	254
22.4. Processing expectations for the refdb DocBook bibliography output.....	254
23. The refdb client/server communication protocol	256
23.1. Prerequisites	256
23.2. Basic principles	256
23.2.1. Message format.....	256

23.2.2. First stage	257
23.2.3. Second stage.....	258
23.3. Commented abstract representation of the client/server protocol	260
23.3.1. First stage	260
23.3.2. Second stage.....	261
23.3.3. Status messages.....	295
Colophon.....	308

List of Tables

5-1. Log level definitions	53
9-1. Sharing extended notes.....	96
10-1. Bibliographic reference types.....	105
11-1. refdbdrc	122
12-1. refdbarc.....	126
13-1. refdbcrc.....	145
14-1. refdbibrc	183
15-1. med2risrc	189
15-2. nmed2risrc	191
15-3. bib2ris exit codes.....	195
15-4. bib2risrc.....	196
15-5. bib2risrc.....	198
15-6. en2risrc	203
15-7. db2risrc.....	205
15-8. marc2risrc	209
16-1. refdbxmlrc	217

Preface

1. refdb in a nutshell

refdb is a free and portable reference and notes database as well as a bibliography application that serves a similar purpose for SGML, XML, and LaTeX documents like Reference Manager (<http://www.refman.com>) or EndNote (<http://www.endnote.com>) do for word processor documents. The bibliography part is something like bibtex for markup languages. refdb consists of an application server which interacts with a relational database on the server side and a variety of clients on the workstation side. These clients perform the common tasks of a reference database, like adding and searching entries, as well as generating bibliographies.

2. Organization of this manual

Part 1 provides a general overview.

Part 2 of this manual explains the installation of the components and the administration of refdb. These tasks should be performed by a system administrator, as you'll need certain privileges that a regular user should not have.

Part 3 explains the general use of refdb for the regular user. This is the place to look if you just need to add or retrieve references or if you want to generate bibliographies.

Part 4 contains all the nasty details of the applications that refdb ships with: All these confusing command-line switches, interactive commands, and configuration file variables that your brain refuses to memorize.

Part 5 is meant for programmers who want to modify or extend the behaviour of refdb. You'll find a few internals which are of no interest for the regular user but might be of interest for people trying to customize or extend refdb.

3. Further information

This document is complemented by the refdb tutorial (<http://refdb.sourceforge.net/doc/tutorial/book1.html>) which explains the tasks a regular user will face in plain English.

Please visit the refdb homepage (<http://refdb.sourceforge.net>) for additional information and new versions.

The refdb-users (<mailto:refdb-users@lists.sourceforge.net>) mailing list is intended to discuss the use of the RefDB package. This covers all topics from installation to running queries. Visit the refdb-users list page (<http://lists.sourceforge.net/lists/listinfo/refdb-users>) to subscribe and for further information.

The `refdb-devel` (<mailto:refdb-devel@lists.sourceforge.net>) mailing list is intended to discuss the development of `refdb`. Use this list to send patches and discuss the further development of this project. Visit the `refdb-devel` list page (<http://lists.sourceforge.net/lists/listinfo/refdb-devel>) to subscribe and for further information.

The `refdb-cvs` (<mailto:refdb-cvs@lists.sourceforge.net>) mailing list receives the `cvs` checkin messages and diffs. Subscribe to this read-only list to be informed about the ongoing development of this project. Visit the `refdb-cvs` list page (<http://lists.sourceforge.net/lists/listinfo/refdb-cvs>) to subscribe and for further information.

Please send general bug reports to the users list (it might be a configuration or usage problem, and would thus be of interest to other users). However, if you have investigated the source code and either have a good idea what might cause the bug or even have a patch, please send the report to the `devel` list. In both cases you may wish to include the output of the **refdb-bug** script. This will create a file `refdb-bug.txt` in the present working directory which contains all sorts of information about your RefDB setup.

`refdb` is maintained by Markus Hoenicka.

I. Overview

Chapter 1. Introduction

1.1. What is refdb?

- refdb is a reference and note database. Use it to manage your collection of offprints of scientific publications, or basically any other collection of printed or electronic documents along with your personal notes.
- refdb is also a bibliography tool for markup languages. The bibliography tool allows to create printable or online documents with formatted citations and bibliographies from DocBook SGML/XML, TEI XML, and LaTeX sources.

1.2. Who should use refdb?

refdb was designed with the needs of a scientist in mind. At least one third of a scientist's work hours are spent reading and evaluating other scientist's publications. In no time you have a pile of offprints or PDF files that you would like to archive in a way that keeps the information accessible. refdb helps you to keep track of those papers by creating a database with the essential bibliographic information, keywords for easy retrieval, your personal comments, the reprint status, a hint where you can find the offprint (or a link to the electronic version), and additional information for your personal categorization.

Extended notes help you organize your references according to topics and allow you to associate additional information with references, keywords, authors, or periodicals. This goes far beyond the one-note per reference approach of commercial tools, although this is of course still supported. Extended notes are associated to the user that created them, but they are visible to all users unless you restrict the display accordingly.

Scientists rarely work alone on a subject. More commonly they are members of a group with similar research interests. It would be a waste of time and effort if each member of the group hunts offprints individually, resulting in duplicate copies, and if each member keeps his personal reference database, making it hard to consolidate an information pool of the group. Instead of selling a complete database engine to each scientist and "allowing" the users to peek at each other's databases, refdb uses the inverse approach: designed as a client/server application from the ground up, it allows a group to maintain a common reference database, but still allows each individual to view only those references that she added or explicitly selected. At the same time, each group member can find out in an instant which offprints are available in the whole group. However, there is no problem if you want to use refdb as an individual user.

Eventually you'll want to publish your own data. You'll have to cite dozens of papers that you've already read. Your refdb database will make it a breeze to retrieve those papers by author, by keyword, or by any other field or combination of fields. If you write your document using one of the supported SGML or XML DTDs or as a LaTeX document, refdb can create the bibliography based on the citations in your document. It will also help you to format the citations and the bibliography according to the instructions

of your publisher. Your paper got rejected and you want to submit it to a different journal? Fear not, refdb can easily re-transform your document and format it for a different publisher.

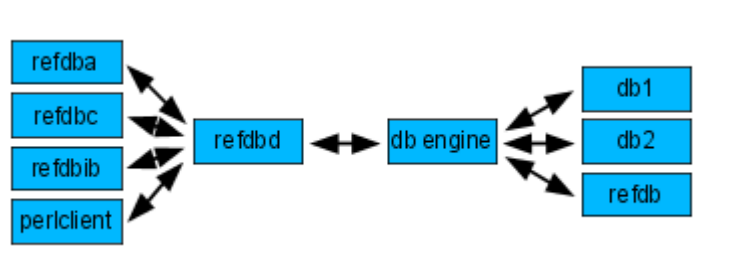
Of course it is also feasible to manage your personal library of contemporary literature. Or your favourite cookie recipes. refdb wasn't precisely designed for these purposes, but it's up to you to make the best use of the package.

If all this sounds useful for you, you should give refdb a try.

1.3. General features

- refdb uses a three-tier architecture (see Figure 1-1), consisting of clients on the workstation side, and an application server as well as the SQL database server on the server side.

Figure 1-1. refdb three-tier architecture



Optionally you may use an internal database engine instead of the external SQL server, resulting in a simpler two-tier setup.

- The data storage proper is done by a SQL database engine. Currently MySQL (<http://www.mysql.com>) and PostgreSQL (<http://www.postgresql.org>) are supported as external database engines. Additionally, an internal SQL engine based on SQLite (<http://www.sqlite.org>) is available, which allows data storage in a single, operating-system and architecture-independent file without any administration overhead.
- The server can run as a daemon in a non-privileged account if security concerns require this. Besides, users can start it as a standalone application on demand. As just about everything is configurable, a thoughtful setup will allow to run several copies of the server on the same box in parallel.
- The refdb clients are no fat applications, but rather a collection of small, portable tools implemented in ANSI C to perform all necessary client-side tasks on any platform with a decent C compiler.
- The refdb clients can be run in an interactive mode or in a batch mode which is useful in scripts.

- Separately available packages integrate reftdb functionality into Emacs and vim, turning these editors into integrated authoring environments for DocBook and TEI documents with easy access to your reference data.
- The adepts of the Perl programming language might be delighted to know that the RefDBClient module allows Perl programs to directly communicate with a reftdb server without using the C clients. This allows the rapid development of custom programs that access RefDB databases.

1.4. Reference management features

- Reference data are imported from text files or from stdin and exported to text files or to stdout. The data model sticks closely to the *RIS* specification that most Windows-based reference managers understand. reftdb natively understands both the tagged RIS format used by other reference management software and an XML format according to the risx.dtd (<http://reftdb.sourceforge.net/dtd/risx/risx.dtd>) similar to RIS.
- Input filters can be used to convert references from various sources to the *RIS* format. Currently reftdb ships with Medline, BibTeX, MARC, and DocBook filters. You are free to use or write any other input filter that you may need. These filters must either create an output file or write the results to stdout for further plumbing. This way, input filters can be written in almost any programming language and it should be easy to extend the list of reference information formats that reftdb can import.

In addition, SGML- or XML-based data formats can be easily transformed to risx using DSSSL or XSLT stylesheets.

- Extended notes are supplied as XML files according to the xnote.dtd (<http://reftdb.sourceforge.net/dtd/xnote/xnote.dtd>).
- The query language is fairly simple yet powerful. You can search in all fields in the database. You can use the Boolean operators *AND*, *OR*, *NOT* to combine search expressions. You can use brackets () to group search expressions. You can use either literal matches or regular expressions in all alphanumeric fields (i.e. most except e.g. the publication year). This gives you enormous flexibility in your search strategies. The readline library reads the user input in all interactive clients. You can recall any previous search strings with a few keystrokes and re-run them or modify them as needed.
- The query results can be displayed in a variety of formats. The standard backends create screen, HTML, XHTML, BiBTeX, DocBook (SGML and XML), TEI, RIS, and risx formats. All output can either be viewed on stdout or with a pager, or the output can be redirected into a file or into a pipe for further manipulation. reftdb provides a simple API to implement custom backends if you need other output formats.

In addition, Perl programmers can use the risx output and the RefDBClient module to easily create any output format they desire.

- reftdb supports all character encodings available on your platform. While the available encodings in the database may be limited by the database engine, reftdb can convert incoming data as well as exported data with only few limitations.

- refdb handles the AV field of the RIS input files in a very flexible way. You can specify a path to a PDF or Postscript version of the document on your harddrive or on the web. The local path can be split into a variable and a static part. The variable part can be specified on the command line e.g. if you access your data remotely via a NFS-mounted share.
- refdb knows the concept of a personal reference list. This feature is useful if a database is shared among several users in a workgroup. In this case, all users benefit from the larger stock of references available in the database. refdb keeps track of the user who added a reference to the database. You can use a switch in the getref command to restrict your search on those references that are associated with your username. On the other hand, if you find out that even your colleagues have one or two interesting papers, you can use the pickref command to add these references to your personal reference list. The personal part of the reference information (the reprint status, the availability, and the notes) are saved for each user individually.

In a similar fashion, the display of extended notes can be limited to the notes of the current user, or all available extended notes are used.

1.5. Bibliography tool features

Bibliography support is currently implemented for DocBook SGML and XML documents, TEI and TEILite XML documents, and for LaTeX documents.

1.5.1. SGML and XML

- Bibliographic output for DocBook works with both the SGML and XML version of the DTD. The DocBook output is sufficiently structured to allow transformation into other SGML or XML document types (this is in fact the way how the TEI bibliographies are created currently). DocBook SGML and XML documents can be transformed with DSSSL and XSL stylesheets, respectively, which act as driver files for the well-known modular stylesheets by Norm Walsh.

The TEI camp has apparently completely switched from SGML to XML, so RefDB supports only XML documents and transformation with XSL stylesheets.

- The implementation of the SGML/XML citations and bibliographies is "non-destructive", i.e. refdb leaves the source document alone and provides all necessary information in a file that can be included into the main document as an external entity. This way, switching from one bibliographic output format to a different one is limited to re-creating this external file without any changes in your source document.

Alternatively you may use a short notation for your citations that makes do with a minimum of markup. The short notation needs an additional (but fully reversible) preprocessing step before bibliographies and formatted output can be created.

- refdb uses "cooked" DocBook bibliography output. This somewhat blurs the distinction between structure and formatting but it is the most efficient way to handle the task of formatting bibliographies

and citations. The TEI bibliography output is even worse as it heavily abuses the all-purpose `seg` element as a wrapper where the elements intended for a particular purpose do not allow a “cooked” use.

1.5.2. LaTeX

- The BibTeX output integrates seamlessly with the LaTeX/BibTeX tools, so there is only one additional command to run on your way from the LaTeX source to the finished document.
- `refdb` creates a BibTeX bibliography file based on the information in a `.aux` file. The latter is created by **latex** from the LaTeX source document. This intermediate bibliography file serves as the input file for **bibtex**. So instead of keeping all your references in a flat text file, `refdb` will create a short bibliography file with the references that your particular document requires.
- `refdb` performs only a very limited amount of formatting for those items which are not well supported in BibTeX (e.g. with BibTeX you need two separate bibliography files if you switch from a format that uses abbreviated journal names to a format that requires the full names). All other formatting is left to the LaTeX/BibTeX system.

1.6. Credits

Throughout the creation of this software the author borrowed code and solutions to particular problems from other software packages in a shameless way. But this is one of the purposes of free software, after all. The following is a hopefully complete list of all programs and libraries that in one way or another helped to get `refdb` on its way.

- `btparse` (<http://www.ctan.org/tex-archive/biblio/bibtex/utis/btOOL/>)
- `expat` (<http://expat.sourceforge.net>)
- `getopt` (<http://www.gnu.org>)
- `libdbi` (<http://libdbi.sourceforge.net>)
- `libdbi-drivers` (<http://libdbi-drivers.sourceforge.net>)
- `libslack` (<http://libslack.org>)
- `MySQL` (<http://www.mysql.com>)
- `ncftp` (<http://www.ncftpd.com/ncftp>)
- `PostgreSQL` (<http://www.postgresql.org>)
- `readline` (<ftp://ftp.gnu.org/pub/readline>)
- `SQLite` (<http://www.sqlite.org>)

- wget (<http://www.gnu.org/software/wget/wget.html>)

Chapter 2. System requirements

2.1. Operating system

refdb was designed to run on as many platforms as possible and has positively been seen running on the following operating systems:

- GNU/Linux
- FreeBSD
- NetBSD
- OSX
- Solaris/SunOS
- Windows/Cygwin

Other Unix-like operating systems may work out of the box or with a minimum amount of tweaking. While this portability excludes fancy GUI bells and whistles, it makes it (most likely) possible to run refdb on the hardware and operating system that you happen to have. Both the clients and the application server should compile on all systems with a decent C compiler like gcc. There should be no problems to run refdb on heterogenous networks. For Windows users the free Cygwin tools are recommended. Cygwin (<http://www.cygwin.com>) is a POSIX layer on top of Win32 including a useful selection of the popular GNU tools (<http://www.gnu.org>).

To simplify the task of porting refdb to other operating systems the package uses autoconf and automake.

2.2. Libraries necessary to build and run refdb

The following libraries are required on your system:

Note: The version numbers in this list do not indicate that it is impossible to build refdb with any lower version number, except where indicated. The numbers just indicate the versions that were used successfully. Lower version numbers *may* work, higher version numbers are assumed to work in all cases.

Some operating systems/distributions use separate packages for the run-time libraries and the development libraries. In order to build refdb from the sources and run the resulting applications, you need *both* types of libraries.

2.2.1. System libraries

These libraries should be offered by all operating systems if they are required on that particular system.

- libreadline (version 4.1 or later)
- libz (version 1.1.3 or later; not required on systems using glibc)
- libnsl, a standard library on Solaris, may be required on this platform.
- libdl, required on all systems that do not include the dl* functions for dealing with dynamic libraries in the standard C library.
- libiconv, required on all systems that do not include the character encoding conversion functions in the standard C library.

2.2.2. Other libraries

- libdbi (<http://libdbi.sourceforge.net>) (version 0.8.x) and libdbi-drivers (<http://libdbi-drivers.sourceforge.net>) (version 0.8.x): a database abstraction layer framework and a set of database-specific drivers.

Note: libdbi provides the framework for the separately available database drivers. refdb currently supports the MySQL, PostgreSQL, SQLite, and SQLite3 drivers. When configuring the libdbi-drivers source package, please make sure to understand that you have to request the drivers you want to build with the `--with-mysql`, `--with-pgsql`, `--with-sqlite`, or `--with-sqlite3` options (you can specify as many drivers as you feel like).

- SQLite (<http://www.sqlite.org>) (version 2.8.11 or 3.0.x): an embedded SQL engine which you can use instead of the external database servers MySQL or PostgreSQL.
- libexpat (<http://expat.sourceforge.net>) (version 1.95.1): a XML parser library.
- btparse (<http://www.ctan.org/tex-archive/biblio/bibtex/utis/btOOL/>) (version 0.34; only required if you want to build the BibTeX import filter): a bibtex parser library.

2.3. Perl modules

Some import filters are implemented as Perl scripts. These scripts require a couple of Perl modules in order to run properly. Some of the modules are available at CPAN (<http://www.cpan.org>), the comprehensive Perl archive network, which allows you to search and download the modules by name. The refdb-specific modules are available as the refdb-perlmod package at the refdb project page (<http://sourceforge.net/projects/refdb>).

marc2ris.pl

- MARC::Record (CPAN)
- MARC::Charset (CPAN)
- refdb-perlmod

med2ris.pl

- XML::Parser (CPAN)
- Text::Iconv (CPAN)
- refdb-perlmod

en2ris.pl

- Text::Iconv (CPAN)
- refdb-perlmod

Note: The **configure** script tests for the existence of the required Perl modules. If your system lacks some of the required scripts, you'll get a warning, but you'll still be able to make and make install refdb. You can install the modules at a later time without rebuilding refdb. In any case, the missing modules will only affect the Perl import filters, but not the core functionality of refdb. That is, if you don't require the Perl scripts, you don't have to install the modules either.

refdb also provides some support for Perl programmers. The RefDBClient module, also available at the refdb project page (<http://sourceforge.net/projects/refdb>), implements Perl classes to directly communicate with refdb. This allows to write custom clients that access the refdb databases and use the whole functionality implemented in refdb. The `test.pl` script shipped with the module gives a basic idea how to write a custom client. See also the chapter about this module in the programmer's manual.

Note: If you're not familiar with Perl modules, here's how almost all of them can be installed: After unpacking the archive, change into the top-level directory of the extracted module. First run **perl Makefile.PL** which will create a system-specific `Makefile`. Then run **make**, optionally followed by **make test** to check the functionality. Finally, run **make install** as root.

2.4. External applications

refdb was designed with the fact in mind that good applications are out there for almost any purpose. Therefore it relies on a few external applications which have to be properly installed. The following list shows the applications which are absolutely required and some applications which will make your life easier.

SQL database server (required for *all* purposes)

refdb uses libdbi (<http://libdbi.sourceforge.net>) as a database abstraction layer to access SQL database servers. refdb currently supports MySQL (<http://www.mysql.com>) and PostgreSQL (<http://www.postgresql.org>) as external database engines. Both database servers run on a wide variety of Unix- and Unix-like operating systems as well as on Windows. Sources and precompiled binaries are available. MySQL should be at least version 3.23.x, PostgreSQL should be 7.1 or later.

If you prefer not to run an external SQL database server, choose the SQLite (<http://www.sqlite.org>) based internal database engine as explained in the libraries section above.

Jade/OpenJade and SP/OSP-based tools (required for DocBook SGML bibliographies, import of DocBook bibliography data)

Jade is a freely available and well-proven DSSSL engine which is based on the SP parser. The Jade package contains a few more SP-based tools, e.g. the nsgmls validator and the sgmlnorm normalizer. refdb uses Jade both to extract the IDs of the references which are cited in SGML documents and to transform SGML documents using DSSSL stylesheets. sgmlnorm is required to preprocess multipart documents using the short notation for refdb citations. The Jade/SP package is available on James Clarks homepage (<http://www.jclark.com>). Prebuilt binaries are available for some platforms, and it builds out of the box on quite a number of platforms. Jade has seen some further development by an independent group of programmers. These newer versions were released as OpenJade/OpenSP and are available at the OpenJade homepage (<http://www.sourceforge.net/projects/openjade/>).

Note: OpenJade has some advantages over Jade for our purposes. If it is possible to obtain or compile OpenJade on your platform, you should go for it. Both Jade and OpenJade can be installed on the same machine without conflicts. The configure script will look for both OpenJade and Jade and will use the former as the default DSSSL engine in the shell script customizations if it is available.

XSLT and FO processors (required for DocBook and TEI XML documents)

If you're working with XML documents and want to transform them using the XSL stylesheets, you'll need some sort of XSL processing machinery. Popular choices are Xalan (<http://xml.apache.org>), Saxon (<http://saxon.sourceforge.net>), and xsltproc (<http://xmlsoft.org/XSLT>). The latter is checked for in the configure script and will be used as the default processor if available. The Java-based tools among these need the Java Virtual Machine

(<http://java.sun.com>) installed, of course. Generating printable output from FO seems to work best with FOP (<http://xml.apache.org/fop/index.html>).

Stylesheets (required for SGML or XML bibliographies)

refdb ships with DSSSL stylesheet driver files for DocBook SGML documents, XSL stylesheet driver files for DocBook XML documents, and XSL stylesheet driver files for TEI XML documents. All of these rely on the respective stylesheets by Norm Walsh (<http://sourceforge.net/projects/docbook>) and Sebastian Rahtz (<http://www.tei-c.org/Stylesheets/>). These must be installed on your system if you want to transform DocBook or TEI documents.

TeX (required for LaTeX bibliographies and recommended for printable output from SGML and XML documents)

Donald Knuth's famous typesetting system is available on almost any platform. You need a TeX system (<http://www.tug.org>) to generate LaTeX bibliographies. It is also convenient to generate nice-looking printable output from SGML and XML documents.

Unix-style text editor (required for all purposes)

This is nothing to worry about if you use some flavour of Unix or Linux: anything that lets you edit texts will do. However, if you have to use Windows, you should be aware that many run-of-the-mill Windows text editors are not suitable for use with refdb as they are not able to handle the Unix-style line endings (LF instead of CR/LF) properly. The standard Windows editors Notepad, WordPad, and MS Word will *not* work unless you use a DOS to Unix conversion filter like dos2unix. These line endings are important in almost all cases, especially when adding and editing references. Some freeware or shareware text-editors for programmers have an option to read and write Unix-style line endings. A few popular Unix text editors are available as Windows ports. Among them are vi clones like vim (<http://www.vim.org>) which is available as a native Windows application and as a Cygwin version, as well as Emacs. GNU Emacs (<http://www.gnu.org/software/emacs/emacs.html>) is available as a native Windows application (<ftp://ftp.gnu.org/gnu/windows/emacs/latest>) and also as a Cygwin package (even for the Cygwin XFree86 port if you wish). XEmacs (<http://www.xemacs.org>) can both be built as a native application and as a Cygwin program (<http://www.xemacs.org/Download/win32/>).

SGML/XML editor (recommended, but not mandatory)

If you plan to create bibliographies, you may need a SGML/XML editor for two reasons. First, the bibliography style specifications have to be written as XML documents (unless someone else has contributed such a file). Second, if you want to create bibliographies for DocBook documents, you want to use a SGML/XML editor anyway. But SGML is plain text after all, so SGML/XML editors make your life easier, but they are not strictly necessary to get something done.

Emacs/XEmacs and the PSGML (<http://sourceforge.net/projects/psgml/>) mode for DTD-based SGML and XML documents or the nxml (<http://www.thaiopensource.com/download/>) mode for RELAX NG-based XML documents are an excellent choice.

Web browser (recommended, but not mandatory)

Most likely you will not have a hard time to find some kind of web browser on your system. A web

browser may be more pleasant to view the query results than the standard output on stdout or a pager. If you use refdb to manage a collection of PDF or Postscript files, the paths to these files will be displayed in the HTML output and you can open the files with one mouseclick.

Perl (import of MARC datasets, import of Pubmed XML datasets, import of EndNote RIS datasets, post-processing BibTeX import files)

The Perl interpreter is available for almost any platform. refdb currently uses Perl for two purposes: the first are the MARC and Pubmed XML import filters, the other is a non-mandatory post-processing step of bibliography data that you import from BibTeX `.bib` files. The MARC import filter requires at least Perl version 5.6.0 and the Perl modules `MARC::Record` and `MARC::Charset`. The Pubmed XML import filter requires the Perl modules `XML::Parser` and `Text::Iconv`. The latter is also required for the Endnote import filter. All things Perl are available at CPAN (<http://www.cpan.org>).

Z39.50 client

In order to retrieve datasets from one of the countless libraries offering their contents through the Z39.50 protocol you'll need a client. The free YAZ toolkit (<http://www.indexdata.dk/yaz/>) includes a sample client which should be sufficient for general use. The toolkit is available as source, RPM and Debian binaries, and as a Windows binary package.

Chapter 3. Current limitations and platform issues

This chapter briefly lists features which are either not yet implemented or which don't work as they should or which work differently on different platforms. These things may be implemented or fixed in future releases, respectively.

3.1. Limitations

- The clients do not use a dynamic buffer to read the data from the application server. This may result in timeouts if you view the results with a pager that can't buffer large amounts of data in the background. To avoid these problems, it's prudent to redirect queries that return large amounts of data into a file.
- The query results can only be sorted by ID (default) and publication year.
- There is no automatic duplicate check for references.
- Support for different character encodings depends on the database server. PostgreSQL offers a variety of different character encodings, including Unicode, which are selectable per database. MySQL versions 4.1 and later support character encodings as well, including Unicode. SQLite offers Unicode support as a compile-time option. SQLite3 uses Unicode by default.
- Export of bibliography styles to BibTeX `.bst` files would be really cool but is not implemented yet.
- There are no manpages yet.

3.2. Platform issues

- Cygwin currently does not support named pipes/FIFOs. `refdbd` uses these to implement a child->parent messaging after the application server has forked. On Cygwin, this messaging is emulated with a temporary file. This is a bit slower (and a lot kludgier and error-prone) than a named pipe and almost certainly is a security hole.

Note: As of this writing, support for named pipes appears to be implemented as an experimental feature in Cygwin. It does not work as expected yet, so as of this writing `refdb` still emulates named pipes on Cygwin.

- Cygwin currently does not implement the detection of pending data on `stdin`. Therefore, data that you send to `stdin` of a client will not be read unless you use the `-f stdin` command line option.

- The Cygwin port of SQLite has a problem working with absolute paths. The safest way is to start refdbd from the root directory (the start script refdbctl does this by default) and use a relative path to the database directory (i.e. leave away the leading slash).
- As the Cygwin library has to translate all calls to Unix system functions to native Windows function calls, there is a small performance penalty in comparison to native Windows or Linux/Unix applications. My general impression however is that this does not affect the overall performance as most time seems to be spent inside the database server. You may gain some milliseconds by running the application server on a Linux/Unix box, though.
- Of course a major problem is the lack of support for more platforms. RefDB *may* work on a variety of platforms besides those mentioned above, but unless you tell me so, I'll never know!

II. Administrator manual

Chapter 4. Installation

This chapter tries to explain the process of the installation. There are some things you should consider before you actually install the software. These will be discussed in the second section. The technical details of the installation on various platforms can be found in the remaining sections of this chapter. But first we start with a brief overview of installing and running refdb for those who never read manuals.

4.1. Quickstart guide

These are the essential steps to get your reference database up and running:

1. Follow the installation instructions for either Unix/Linux or Windows.
2. Finish the installation by creating and filling the internal database.
3. Start the application server refdbd, either manually or as a daemon, as explained in the OS-specific installation sections.
4. Create one or more reference databases.
5. Add one or more users that are permitted to work with this database (not required if you use the SQLite database engine as it lacks access control).
6. Add bibliography styles.

Now it is up to the users to populate the database with references, run their queries, create bibliographies, and whatever they may desire.

4.2. Upgrading from an older version

If you already run an older version of refdb and want to upgrade to the latest version, you may have to change a few things in your local setup in order to reuse existing databases and existing configuration files. This section lists all backwards-incompatible changes of the past few versions (the complete list since the days of yore can be found in the file `README`).

Note: The file `UPGRADING` contains instructions how to avoid losing your data in the case of non-trivial changes in the new version.

4.2.1. 0.9.5

- Once again, the database schema of the main database has changed. However, starting from this version the name of the main database is a configure option. If you do not have to run this and an older

version in parallel, you don't have to change the configuration. However, if you want to keep your existing setup while test-driving the new version, there is a new configure switch `--with-main-db` to set the name of the main database.

In the first case, you'll have to replace your existing main database with the new version. In the second case, you create another main database with a different name. Only the new version of `refdb` will be able to access this database if configured properly.

Finally, a bug in the MySQL table definitions was fixed. If you're using MySQL as your database engine, you'll have to migrate your reference databases as well. 0.9.5 provides a SQL script that changes the table definitions appropriately.

- The citation and bibliography style DTD (`citeststyle.dtd`) was extended to facilitate the formatting of bibliographies in terms of indentation and font sizes. Styles written with older versions of the DTD will continue to work with `refdb` 0.9.5 or later. However, if you want to use the new features, you should extend the style files according to the new DTD and add the styles again.

4.2.2. 0.9.4

- In order to support the new link elements, the database schema of the common database has changed. The name of the database was therefore changed back to `refdb`. This will allow to run 0.9.4 and older versions in parallel if desired.
- The support for extended notes required substantial changes in the database schema of the reference databases. It is required to export all existing references to RIS files and reimport them into new databases after upgrading to 0.9.4:
 1. Make all sorts of backups of your existing databases. Use the native tools of your database server to create SQL dumps. In case of MySQL or SQLite databases, you can also make copies of the subdirectories and files, respectively.
 2. While still running your pre-0.9.4 installation, use the `refdbc` command **`getref -t ris -o dbname.ris :ID: >0`** to dump your databases to RIS files.
 3. After installing and starting 0.9.4, create a new database and read back your existing data with the `refdbc` command **`addref -t ris -d newdbname dbname.ris`**

4.2.3. 0.9.3

The citation key code had some serious flaws in previous versions. In order to work around these flaws it is recommended to re-create existing databases from a `risx` dump. This will fix the following issues:

- In order to work properly in SGML documents, the citation keys must be all uppercase. Mixed case is fine if you use only XML or BibTeX.

- In order to avoid troubles with incompatible character encodings, the citation keys should be restricted to the 7-bit ASCII range.

The following procedure is recommended to re-create your databases:

1. Make all sorts of backups of your existing databases. Use the native tools of your database server to create SQL dumps. In case of MySQL or SQLite databases, you can also make copies of the subdirectories and files, respectively.
2. After upgrading to 0.9.3, use the `refdbc` command **getref -t risx -o dbname.xml :ID:>0** to dump your databases to XML files.
3. Decide whether you want to use your setup for SGML documents. If you will use SGML documents, the citation keys have to be all uppercase. You can do this manually, use a Perl one-liner, or configure `refdbd` to uppercase citation keys when adding datasets. Use either the `-U` command line switch or set the `upper_citekey` configuration variable appropriately. You can of course process XML and LaTeX documents with these settings as well, but you'll have to enter the citation keys in their uppercase forms in XML and LaTeX documents.
4. Create a new database and read back your existing data with the `refdbc` command **addref -t risx newdbname.xml**

4.2.4. 0.9

- The schema of the common database `refdb` has changed. To allow old and new `refdb` versions to run in parallel, versions 0.9 and later use `refdb1` as the common database. Create this database after installation of the `refdb` components according to the user manual.
- The `refdbvar` database used by `refdb` versions prior to 0.9 is obsolete. After upgrading to 0.9 you may safely drop this database.
- The schema of the reference databases has changed. The main table `t_refdb` now has an additional column for the citation keys. In order to use your existing reference databases, please follow the instructions in the file `UPGRADING`.
- The configuration files are now expected to be in a subdirectory `refdb` of the system configuration directory. The example configuration files will be installed in this new location. To reuse your existing configuration files, please move these to the new location and add entries for variables new in 0.9.
- The `remoteadmin` configuration variable for `refdbd` now uses `'t'` and `'f'` as argument instead of `'1'` and `'0'`.
- The `cgiurl` configuration variable for `refdbc` was renamed to `cssurl` which better reflects its purpose (ok, it was a braino to call it `cgiurl` in the first place).

4.3. Things to know before you start

4.3.1. Which database server?

refdb currently supports MySQL and PostgreSQL as external database servers as well as SQLite as an embedded database engine. This section tries to help you decide which one to pick.

The first issue is whether you want to run an external database server or not. External database servers scale better if many users share databases and they provide access control. The external database servers also use more fine-grained locking mechanisms which allow concurrent read and write accesses, whereas the SQLite engine will lock the entire database for write accesses. However, the latter does not provide access control and thus doesn't require any sort of user administration.

Rule #1. If you don't intend to share databases, or if running a database server scares you in any way, then you may better off with SQLite.

Another issue is the way how the database engines store their data. SQLite is unique in that it uses a single architecture-independent file per database which makes transferring the data to a different box a breeze. The external database engines use more sophisticated ways to organize their data, but you need some basic administrative skills in order to replicate the data.

Rule #2. If you cannot rely on remote access to your databases (something which refdb is well suited for) but have to take your data physically with you while travelling, SQLite is a better choice.

Now some words about the external database servers. As with many other fundamental schisms in the Unix world (vi vs. Emacs, KDE vs. Gnome, to name a few), both database servers supported by refdb have followers which are semi-religious about their choice. Both MySQL and PostgreSQL are robust and well-proven. This leads us to:

Rule #3. If you already use one of the servers, then by all means use it also for refdb. Being familiar with the server and having it happily running usually outweighs any advantages that the other server might have.

But what if you do not yet run a suitable database server? You can browse the web and read for hours about the differences between MySQL and PostgreSQL, but for the purpose of managing refdb reference databases it boils down to one essential difference: MySQL is faster.

This leads us to:

Rule #4. If you cherish speed over anything else, use MySQL.

Note: There's a few more differences that you should be aware of: PostgreSQL has transaction support by default. MySQL supports transactions only if you use InnoDB tables. If you want this additional peace of mind from MySQL, make sure InnoDB is the default table type. SQLite does not support Unix-style regular expressions. If you'd like to use these more versatile expressions instead of the simpler SQL regular expressions supported by SQLite, choose MySQL or PostgreSQL.

4.3.2. Where do the components go?

As refdb is a three-tiered client-server application, you have considerable freedom to distribute the components among your computers. Although refdb shines in a network environment, there is absolutely no problem to run all components on a single standalone workstation.

Note: Please keep in mind that there's one tier less if you choose the SQLite embedded database engine. The databases will always be on the filesystem of the machine that runs refdbd (this doesn't exclude putting the files on an NFS share if you have a good reason to do so).

The basic idea of the client-server model has several implications:

- Many workstations can access a single server running the database server. Thus many people can access the same databases without the pain of duplicating the data and the database engine on every single machine.
- A considerable part of the computing effort is done outside of the workstations. Therefore even rather lame workstations may be sufficient to access and manipulate the data. The database server should run on a decent machine, though (better not that dusty 486 that has doubled as a paperweight since 1990).
- Updates of the software will mainly affect the database server and the application server. This considerably reduces your workload, as the workstations need to be updated less frequently.

The most common scenarios for using refdb will be on a department or institute network and on a standalone workstation. Let's see how these scenarios differ:

4.3.2.1. Installation on a standalone workstation

This is obviously the simplest case. The clients, the application server, the database server, and the databases reside on the same physical machine (see Figure 4-1). The only requirement for the workstation is that a TCP/IP network is installed. This is necessary as the three layers of refdb always communicate via TCP/IP sockets. The IP address 127.0.0.1 has to be specified in the configuration files of the clients and of the application server.

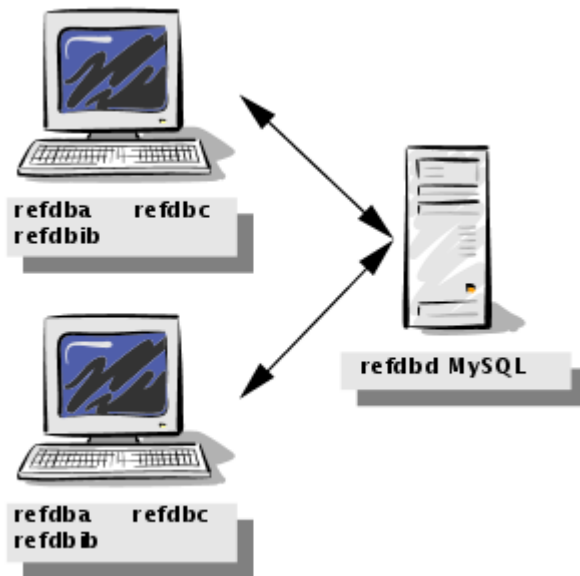
Figure 4-1. refdb on a standalone workstation

4.3.2.2. Installation in a network

In a network you can take advantage of the client-server model and distribute the workload between your computers. Although the three layers can well be distributed between three physical machines, it may be more useful to install the application server on the same machine as the database server and the databases (see Figure 4-2). A dedicated or general-purpose server may be most suitable to hold these components, as a workstation may get sluggish if it has to answer a lot of database requests.

The clients as well as scripts and support files have to be installed on all workstations that will be used to access the databases. The client for administrative tasks, `refdba`, can be restricted to the workstations of system administrators or otherwise experienced staff.

Figure 4-2. refdb on a network



4.3.3. The mystery of the configuration files

Like with most Unix-style software packages, the behaviour of the refdb applications can be tweaked by configuration files. Wherever it makes sense, there is one global config file with useful admin-picked defaults, and another user config file for the individual user to play with. The purpose of the configuration files is to set some reasonable default values for the command-line switches of the refdb programs. Once you have set these, you will never have to specify these values on the command line again, unless you want to temporarily override them.

4.3.3.1. Types of configuration files

All refdb applications and scripts that use configuration files (these are the server refdbd, the clients refdbc, refdbib, refdba, the script refdbxml, as well as the conversion filters bib2ris, db2ris, med2ris.pl, marc2ris.pl, and en2ris.pl) can use two configuration files each. One global configuration file is supplied by the system administrator and can be used to set values that are common for all users on that box, like the IP address of the application server. Another file can be used by every user to supply the values that were not set in the global file or to override settings in this file. The users' copies can have a leading dot to hide the files (the refdb programs will first try to read a hidden configuration file, and only if that cannot be found they try to read a non-hidden file).

refdbc, bib2ris, marc2ris.pl, and med2ris.pl use a second global configuration file if they are run as a CGI applications. A local configuration file does not make sense in this case.

The default location for the global configuration files is `/usr/local/etc/refdb`. There are two ways to change this. If you compile refdb from the sources you can specify a different directory with the `--prefix` or `--sysconfdir` options of `./configure`. E.g. if you specify `--sysconfdir=/etc`, then the configuration files will be installed in `/etc/refdb` (the `refdb` part is automatically appended by the refdb install routines). If you use precompiled binaries, use the `-y` command line option to specify the directory. In this case you have to specify the full path, i.e. `/etc/refdb` to read the configuration files installed by the previous example.

The user copies of the client configuration files are expected to be in the users' home directories as specified by the environment variable `HOME`.

4.3.3.2. Configuration file syntax

All configuration files share a common syntax. There are just three essential things to know:

- All information is stored as pairs of whitespace-separated items, one pair on each line. The first item on the line specifies the variable name, the second item specifies the variable value. Whitespace means one or more spaces or tabs in any combination.
- Everything to the right of a hash sign (`#`) is a comment. The rest of the line is ignored.
- The line endings are Unix-style (`0x10`, not DOS-style `0x13 0x10`), regardless of the operating system.

4.3.3.3. A configuration example

The whole configuration stuff may sound a bit confusing, so let us now look at a simple configuration example that illustrates the principles laid out above.

The following is a listing of `/usr/local/etc/refdb/refdbcrc`, our global refdbc configuration file in this example:

```
# This is the global configuration file for refdbc
serverip 127.0.0.1
port 9734
pager more
timeout 180
# end of refdbcrc
```

This is the corresponding copy that one of the users of the system created as `/home/joe/.refdbcrc`:

```
# This is the user configuration file for refdbc
pager less
username joesixpack
passwd *
timeout 30
```

```
# end of .refdbcrc
```

As you can see our hypothetical system administrator configured the IP address (`serverip`) and the `port` where `refdbd` listens to the client requests. This value is most likely the same for all users on the system, so this is nothing to worry about for the users. `more` is defined as the default `pager`, and the `timeout` is set to 3 minutes.

Joe Sixpack, our reckless user, does not like `more` as a pager and prefers to use `less` instead. He also thinks that half a minute as a timeout should be enough. Both of these settings override the corresponding values in the global file. `serverip` and `port` are not redefined in the user's copy, so the values of the global file take effect. Joe also defined `username` (which happens to be different from his login name "joe") and `passwd` so the correct values will be used for the database access (the asterisk in the `passwd` field will cause `refdbc` to ask for the password interactively for security reasons).

4.3.3.4. Configuration file variables

For a listing of available configuration file variables please see the tables for `refdba`, `refdbc`, `refdbib`, `refdbd`, `refdbxml`, `bib2ris`, `db2ris`, `med2ris.pl`, `marc2ris.pl`, and `en2ris.pl`.

4.3.4. Environment variables

`refdb` uses the following environment variables to locate the files and directories it needs to run properly.

HOME

This variable should be set for all users anyway. It is used to locate the personal configuration files for the `refdb` clients.

SGML_CATALOG_FILES

If you process SGML files, this variable will be consulted to locate the catalog files required for resolving public identifiers to their local filename equivalents.

Note: On some systems, the package system maintains a master catalog whose path is hard-coded into the SGML applications. In this case, the variable is not required.

XML_CATALOG_FILES

If you process XML files, this variable may be consulted to locate XML catalogs. If this variable is not set, many tools look into the default location `/etc/xml/catalog` instead. Remember that some XSLT processors need access to additional Java classes to provide XML catalog support at all.

4.3.5. Some notes on the filesystem

The default installation procedure will install the refdb files in locations compatible with the filesystem hierarchy standard. You will learn in the following sections how to change where the refdb files will be installed if you want to adapt the installation to specific needs of your system. To get a better idea of what you have to take care of if you don't like the defaults, here is a list of the directories used by refdb:

/usr/local/bin

This directory will receive all binary files and shell scripts.

/usr/local/etc/refdb

All global refdb configuration files end up in this directory.

/usr/local/share/refdb

This directory contains shareable, operating system independent files. The files are organized in a couple of subdirectories:

- `css` contains a cascading stylesheet suitable for the HTML output of the **getref** command.
- `declarations` contains the default SGML declarations.
- `dsssl` contains DSSSL stylesheets.
- `dtd` contains the document type definitions used by refdb.
- `examples` contains a few example reference data files as well as SGML and XML test documents using refdb citations.
- `sql` contains SQL scripts used to initialize databases.
- `styles` contains some XML files containing bibliography styles.
- `xsl` contains XSLT stylesheets.

/var/db/refdb

holds the database files of embedded database engines and a version file for use by package installation scripts

4.4. Installation on Linux and other Unix variants

refdb was designed with portability in mind. Thanks to the GNU autotools installation on many Unix variants is straightforward, others need only a little tweaking. The following instructions provide a guideline for all Unix-like systems. Please see also the hints for specific operating systems at the end of this section.

4.4.1. Generic instructions

4.4.1.1. Prerequisites

These instructions assume that the following software is installed and functional on your computer or on your network before you start:

- A SQL database engine. For an external database server, choose between MySQL (<http://www.mysql.com>) (version 3.23 or later) or PostgreSQL (<http://www.postgresql.org>) (version 7.1 or later). The database server does not have to physically run on the box(es) where you plan to install refdb, it is sufficient if it is accessible through the network.

If you prefer an embedded database engine, please use SQLite (<http://www.sqlite.org>) instead.

- As far as non-standard libraries and header files are concerned, you'll need the readline library and headers (available at the GNU FTP site (<ftp://ftp.gnu.org/readline/>)). Database access requires the libdbi library (<http://sourceforge.net/projects/libdbi>) and headers as well as the libdbi-drivers (<http://sourceforge.net/projects/libdbi-drivers>) package. Furthermore, you'll need the expat (<http://sourceforge.net/projects/expat>) library. If you need the BibTeX import filter, you'll also want to have the btparse (<http://www.ctan.org/tex-archive/biblio/bibtex/utis/btOOL/>) library.

Note: Some operating systems/distributions use separate packages for the run-time libraries (usually `.so` files) and for the development libraries (usually `.a` files). In order to build and run refdb, you need both packages for each library.

- If you want to import Pubmed or MARC datasets, please get the refdb-perlmod (<http://sourceforge.net/projects/refdb>) package. This collection of Perl modules is required to run the Pubmed and MARC import filters shipped with refdb. These Perl modules in turn depend on MARC::Record, MARC::Charset, XML::Parser, and Text::Iconv, available at CPAN (<http://www.cpan.org>).

4.4.1.2. Installation from the sources

Note: The installation steps proper (as opposed to the build steps) should be run with root privileges.

1. Unpack the archive in a convenient directory: **tar -xzf refdb-x.y.z.tar.gz** (the actual filename depends on the version).
2. If you do not want to build in the source directory, create an empty build directory.
3. `cd` into the new `refdb-x.y.z` source directory or into your separate build directory

4. Use `./configure --help` to see a list of things you can customize. If you use a separate build directory, use the relative path to `configure` in the source directory. Some important options are:

Note: All paths and URLs in the following options should be entered without a trailing slash.

--prefix=PREFIX

By default, all files will be installed in the `/usr/local` tree. Use this option to use a different install root, e.g. `/usr` or `/opt`.

--datadir=DIR

The data files will be installed in `/usr/local/share/refdb` unless you use this option. The data will be installed in the directory `DIR/refdb`. That is, specifying `"--datadir=/usr/local/share"` is equivalent to the default behaviour. The configuration variable `refdblib` (which will be automatically generated in the example configuration files) must point to the `refdb` data directory.

--sysconfdir=DIR

The global configuration files will be installed in `/usr/local/etc/refdb` unless you specify a different directory here.

--with-libdbi-lib=DIR

Use this option to specify the directory that contains the `libdbi` library if it is not in the default library path.

--with-expat-lib=DIR

Use this option to specify the directory that contains the `expat` library if it is not in the default library path.

--with-btparse-lib=DIR

Use this option to specify the directory that contains the `btparse` library if it is not in the default library path.

--with-sgml-declaration=PATH

Specify the full path to a suitable SGML declaration for your SGML files. If this option is not used, `refdb` will use its own copy of `docbook.dcl` stolen from the DocBook DTD distribution. This SGML declaration also works for a variety of other DTDs.

--with-xml-declaration=PATH

Specify the full path to `xml.dcl` which is the SGML declaration for XML files. If this option is not used, `refdb` will use its own copy of `xml.dcl` which should work just fine.

--with-docbook-xsl=PATH

Specify the full path to the root directory of the DocBook XSL stylesheets.

Note: This option is required only on systems that do not maintain XML catalogs. If your system is set up properly to resolve public identifiers like those in the XSL stylesheets by XML catalogs, leave out this option. **configure** checks whether the required stylesheets are accessible, so watch out for error messages. If the stylesheets can't be found, either install them, fix your catalogs, or use this option to hardcode the path.

--with-tei-xsl=PATH

Specify the full path to the root directory of the TEI XSL stylesheets.

Note: The TEI Consortium (<http://www.tei-c.org>) ships `tei-xsl-5.2.9.zip` (http://sourceforge.net/project/showfiles.php?group_id=106328&package_id=141124) which contains the stylesheets for both p4 and p5. The root directory which you want to specify here is the directory which contains the p4 and p5 subdirectories, e.g.
`/usr/local/share/xsl/tei-xsl-5.2.9.`

--with-classpath-root=PATH

The `refdbxml` script assumes that all Java classes for the Java parsers and xslt engines are stored in a class repository, i.e. all in the same directory. Specify this directory with this option. If you keep the relevant Java classes in different directories, either create symlinks or customize `refdbxml` manually.

--with-refdb-url=URL

The HTML files and templates for the web interface require absolute URLs matching your web server setup. Specify the URL of the directory that contains the `refdb` HTML files. If you follow the Apache setup instructions in this manual, this will be a string like `"http://www.mycomp.com/refdb"`, where `"www.mycomp.com"` is the hostname of your computer running Apache and `"refdb"` is the directory alias used for the `refdb` web pages.

Note: This URL affects the links in the HTML files and templates. A wrong or missing value of this URL will break the navigation as well as the access to stylesheets and javascript files.

--with-var-dir=PATH

Use this option to specify a directory where refdbd can write its PID file (a file containing the process ID). By default, `/var/run` will be used.

--with-log-dir=PATH

Use this option to specify a directory where refdb programs can write log files to, if logging is directed to a custom file. By default, `/var/log` will be used.

--with-db-server=DBS

By default, all configuration files are preconfigured for use with the MySQL database server. Use this option to preconfigure the configuration files for a different database server (currently `mysql`, `pgsql`, and `sqlite` are recognized as values for `DBS`).

Note: This option does not modify the binaries that you will create. These will work with all supported database engines regardless of what you select here. The only things affected by this option are a few settings in the `refdbdrc` configuration file. You can change them after the installation anytime.

--with-main-db=dbname

refdb uses one main database to store citation styles and other stuff. There is exactly one such database per installation with the default name `refdb`. The only reason to change this name is if you want to run this and an older, incompatible version of refdb in parallel.

Some of these options are used to customize the shell scripts, XSL stylesheets, example configuration files, and HTML files which are part of refdb. If you do not specify these options now, you can still build and install the package, but you will have to customize the scripts and stylesheets manually in order to make them work. Doing it now is *much* easier. You've been warned.

Start the configuration with the command `./configure`, specifying any additional options as you may need. Use the relative path to `configure` in the source directory if you build in a separate directory.

Note: If your system keeps non-system header files in odd places, it may be necessary to set the `CFLAGS` environment variable before you run `configure`. E.g. if headers like `expat.h` are stored in `/usr/local/include`, you should run `CFLAGS="-I/usr/local/include" ./configure` instead, specifying additional options as necessary.

5. make

Note: The autotools-generated Makefiles apparently prefer (or require?) GNU make. If `make` results in spurious error messages *about the Makefile syntax*, try to run `gmake` instead as your regular `make` is apparently not the GNU version.

6. **make install**

This will install the binaries and scripts in `/usr/local/bin` and the data in `/usr/local/share/refdb` unless you chose different directories in the configure step. Again, run **gmake install** instead if your regular `make` is not the GNU version.

7. To finish the installation, please follow the instructions in the section *Finishing the refdb installation* below.

If this procedure results in strange error messages, you probably use a platform that is not supported yet. The author appreciates a porting effort or a description of the problem (in this particular order).

4.4.1.3. Living on the bleeding edge: installing a CVS version

While the official release versions of `refdb` are your best bet to get a stable installation, you may be interested to test the latest developments "in statu nascendi". The CVS version is guaranteed to compile on the author's development platform (currently FreeBSD 5.4), but may fail on other platforms. Known bugs may have been fixed, but new ones may have crept in as well. New features may be implemented, but might not be without problems. You've been warned. If you still want to go ahead, this is what you need to do:

1. Change into a suitable directory. CVS will create a subdirectory called `refdb` during the following steps
2. Run the following command (adapt this accordingly if you use some graphical CVS frontend instead of the command line tool):

```
$~/build cvs -d :pserver:anonymous@cvs.sourceforge.net:/cvsroot/refdb login
```

Just press enter if you're asked for a password. This login is required only once as CVS will remember the required information for all later accesses.

3. Check out the latest `refdb` sources with this command:

```
$~/build cvs -d :pserver:anonymous@cvs.sourceforge.net:/cvsroot/refdb -z3 checkout refdb
```

This will retrieve the HEAD code which is pretty much experimental and unlikely to work at the moment. The more usable stable branch, which receives most of the current development work, can be retrieved with this command:

```
$~/build cvs -d :pserver:anonymous@cvs.sourceforge.net:/cvsroot/refdb -z3 checkout -r Release_0_9_
```

Other interesting modules to check out are `perlmod` (the Perl modules and the Perl client library), `makestyle` (the Perl script to write bibliography styles from scratch), and `elisp` (the Emacs support files).

If you are interested in a particular release or branch, use a command along the lines of this example instead:

```
$~/build cvs -d :pserver:anonymous@cvs.sourceforge.net:/cvsroot/refdb -z3 checkout -r Release_0_9_
```

4. If you want to keep track of the latest developments, you'll have to update your CVS version once in a while. Run this command from within the top-level source directory:

```
$~/build/refdb cvs update
```

Consider subscribing to the refdb-cvs mailing list as you'll be notified about each cvs checkin.

5. The CVS version lacks all autotool-generated files. You'll have to create them with your local autotools. refdb contains a small script called `autogen.sh` in its top-level directory which performs all necessary steps.

```
$~/build/refdb ./autogen.sh
```

Note: The autotools can cause severe headaches at times. If the above command causes errors, read `autogen.sh` and run the commands manually, substituting different versions as appropriate. E.g. substitute **automake** with **automake-1.8** if this gives better results. The native autotools shipped with FreeBSD 5.4 are unusable for our purposes, you have to install the gnu-autotools package instead. You have to modify your PATH to pick up the latter, and fiddle with the m4 macro paths in order to see the light.

6. The CVS version also lacks a few other things which are included in the source tarballs. To rectify this, you may have to use the following additional **./configure** switches:

--with-trang-jar=PATH

The CVS sources contain the DTDs from which you have to build the RelaxNG schema files if you need them. Pass the full path of `trang.jar` to this option. If not specified, the `.rng` files will be mere copies of the DTD files.

--enable-docs

The refdb manual is not built by default. Use this switch to generate the documentation from the sources. In order to successfully build the docs, you need the following tools installed:

- OpenJade (<http://www.sourceforge.net/projects/openjade/>): the DSSSL engine used to transform the sources
- JadeTeX (<http://sourceforge.net/projects/jadetex/>): a TeX macro package used to generate printable output
- TeX: the typesetting engine used to generate printable output
- Batik SVG Rasterizer (<http://xml.apache.org/batik/svgrasterizer.html>): an image converter used to create the images from the SVG sources. The Makefile calls a script in your path called `batik-rasterizer` which is supposed to run this Java application appropriately.
- dtdparse (<http://dtdparse.sourceforge.net>): A Perl script (which in turn depends on a few Perl modules, see the installation instructions) which generates the DTD documentation.

7. Finally you're ready to build refdb like you would with a released version.

There is also a neat shell script available on the refdb download page (<http://refdb.sourceforge.net/download.html>) which helps you to keep a CVS version up to date with a minimum number of keystrokes.

4.4.1.4. Installation of reftdb as a daemon

This is all it takes to run reftdb from the command line. If you want to start reftdb as a daemon at system startup, a few more steps are necessary. The exact procedure varies greatly between operating systems and distributions. First we'll look at SysV-style systems (most Linux distributions), then at BSD-style systems (BSD-derived Unices and the Slackware Linux distribution).

4.4.1.4.1. SysV-style

The following procedure describes the setup on a Debian GNU/Linux 2.2 system. With a little help of your system handbook you should be able to adapt this to your system. On many systems `/etc/init.d/README` contains just what you need to know or at least it points you to the correct resources.

1. Review the parameters in the script `refdb` in the `scripts` directory of the source distribution (this file is not automatically installed). If necessary, change the paths and names to your needs and adapt the following steps. Make sure the `BSDSTYLE` variable is set to "NO" (this is the default value).
2. Copy the script `refdb` to `/etc/init.d/` and make sure it is executable: **`chmod 755 /etc/init.d/refdb`**.
3. Create symbolic links from every runlevel directory that should start reftdb to `/etc/init.d/refdb`, e.g. **`ln -s /etc/init.d/refdb /etc/rc2.d/S93refdb`**. The numbers in the link names are a convenient and simple way to determine the sequence of daemon starts. As most likely no other daemons rely on reftdb, you can choose as high a number as you want (the daemons are started in lexicographical order).
4. In analogy to the previous step, generate symbolic links in every runlevel directory that should stop the daemon. Usually these are the runlevels 0 (system halt) and 6 (reboot):
`ln -s /etc/init.d/refdb /etc/rc0.d/K20refdb`
`ln -s /etc/init.d/refdb /etc/rc6.d/K20refdb`
5. Review the settings in the script `refdbctl` which by default is installed in `/usr/local/bin/`. Most likely the script was properly customized for your system in the build step so you don't have to change anything.

When you boot, halt, or reboot the system you should see messages on the screen telling you that the daemon has been started or stopped successfully. If you don't, please check again all paths in the scripts, the file permissions, and the runlevels you're looking at. Try to run the control script from the command line, e.g. **`refdbctl start`**, to distinguish between general setup problems and init-related setup problems.

4.4.1.4.2. BSD-style

The following procedure describes the installation on a FreeBSD 4.3 system. Other systems might differ somewhat, but you should get the idea.

1. Review the parameters in the script `refdb` in the `scripts` directory of the source distribution (this file is not automatically installed). Set the value of the script variable `BSDSTYLE` to "YES" (the sole purpose of this variable is to make the boot message blend in seamlessly, if you happen to care for aesthetics). If necessary, change the paths and names to your needs and adapt the following steps.
2. Copy the script `refdb` to `/usr/local/etc/rc.d/refdb.sh` and make sure it is executable:
`chmod 755 /usr/local/etc/rc.d/refdb.sh`.

Note: The suffix `.sh` is mandatory on BSD-style systems.

3. Review the settings in the script `refdbctl` which by default is installed in `/usr/local/bin/`. Most likely the script was properly customized for your system in the build step so you don't have to change anything.

When you boot, halt, or reboot the system you should see messages on the screen telling you that the daemon has been started or stopped. If you don't, please check again all paths in the scripts and the file permissions. Try to run the control script from the command line, e.g. **`refdbctl start`**, to distinguish between general setup problems and system-related setup problems.

4.4.2. OS-specific hints

This section contains a few hints about the installation on some popular Unix-like operating systems.

4.4.2.1. Linux

The generic instructions should work out of the box for most if not all Linux distributions.

4.4.2.1.1. Debian GNU/Linux

Debian (<http://www.debian.org>) keeps all configuration files in `/etc` and subdirectories thereof which makes backing up the configuration of the whole box a breeze. To support this nice feature, you want to add `--sysconfdir=/etc` to your configure options. The configuration files will then be in `/etc/refdb`. The following is an example of a complete configure call that works on Debian 3.0 (the line was split into several lines for the sake of clarity; type everything on one line and skip the backslashes):

```
#~ ./configure --sysconfdir=/etc \
--with-docbook-xsl=/usr/share/sgml/docbook/stylesheet/xsl/nwalsh \
```

```
--with-tei-xsl=/usr/local/lib/sgml/stylesheet/xsl \
--with-refdb-url=http://localhost/refdb
```

4.4.2.2. FreeBSD

The BSD version of **make** cannot handle the autotools-generated Makefiles correctly. Use **gmake** instead, and set **CFLAGS=-I/usr/local/include** during the configuration step. The full configure command, using the default installation paths for the accessory files and programs, looks like this:

```
#~ CFLAGS=-I/usr/local/include ./configure --with-expat-lib=/usr/local/lib \
--with-classpath-root=/usr/local/share/java/classes --with-refdb-url=http://localhost/refdb
```

The shared data end up in `/usr/local/share/refdb`, the configuration files will be in `/usr/local/etc/refdb`.

4.4.2.3. NetBSD

The BSD version of **make** cannot handle the autotools-generated Makefiles correctly. Use **gmake** instead, and set the following environment variables before running **./configure**:

```
#~ setenv CFLAGS "-I/usr/pkg/include -I/usr/local/include -L/usr/pkg/lib"
```

```
#~ setenv LDFLAGS "-L/usr/pkg/lib -R/usr/pkg/lib -R/usr/pkg/lib/mysql"
```

```
#~ ./configure --with-mysqclient-lib=/usr/pkg/lib/mysql --with-btparse-lib=/usr/local/lib --with-refdb-url=http://localhost/refdb
```

Use additional options as needed. The shared data will end up in `/usr/pkg/share/refdb`, the configuration files go to `/usr/pkg/etc/refdb`.

4.4.2.4. Solaris

Both the standard C compiler and the standard make program will not give the desired results when building `refdb`. Use `gcc` and `gmake` instead.

4.4.2.5. OSX

Use `fink` (<http://fink.sourceforge.net>) to conveniently install all the additional software needed. Then help **configure** find the fink-installed packages in `/sw` by setting the following environment variables:
CFLAGS="-I/sw/include" LDFLAGS="-L/sw/lib" ./configure [your_options].

4.5. Installation on Windows NT/2000/XP

4.5.1. Prerequisites

In order to run refdb you'll need a few external apps. The most important thing are the Cygwin tools which provide a Unix-like environment on the otherwise incompatible Windows platform. Additionally you'll have to build or obtain prebuilt versions of a few libraries which are not part of the standard Cygwin distribution (unless you use prebuilt binaries that contain all non-standard libraries).

4.5.1.1. External applications

- Cygwin toolkit (<http://www.cygwin.com>), DLL version 1.5.0 or later (the current version as of this writing, 1.5.12, has serious problems with PostgreSQL, though). The Cygwin distribution uses individual packages for the various parts. In addition to what is installed automatically (the base distribution), please make sure to select the following packages: libxml2, libxslt, expat, Perl, and a full TeX installation if you wish to create Postscript or PDF files.
- A SQL database engine. Currently you can choose between MySQL (<http://www.mysql.com>) (version 3.23 or later) and PostgreSQL (<http://www.postgresql.org>) (version 7.1 or later). The database server does not have to physically run on the box(es) where you plan to install refdb, it is sufficient if it is accessible through the network.

Cygwin offers a prepackaged version of PostgreSQL (7.3.3 as of this writing) which used to be the most convenient way to run a database server on your Cygwin box, but as stated above it may be broke when you try. Please follow the instructions in the readme shipped with the package to install and initialize the database server. It is crucial that you install and run the separately available cygipc package, available only here (<http://www.neuro.gatech.edu/users/cwilson/cygutils/cygipc/index.html>).

SQLite is not available as a package, but current versions build straight out of the tarball and run with one minor annoyance (absolute paths to database files don't work). MySQL is currently not officially supported on this platform, but you may still get it to work.

- If you want to import Pubmed or MARC datasets, please make sure to install the Perl interpreter available for Cygwin and get the refdb-perlmod (<http://sourceforge.net/projects/refdb>) package. This collection of Perl modules is required to run the Pubmed and MARC import filters shipped with refdb. These Perl modules in turn depend on MARC::Record, MARC::Charset, XML::Parser, available at CPAN (<http://www.cpan.org>).

The mentioned websites offer plenty of support as web documents or with mailing lists, so it should be possible to figure out how to do a basic install for these suites.

4.5.1.2. Libraries

- The libdbi library (<http://sourceforge.net/projects/libdbi>) and headers, along with the libdbi-drivers (<http://sourceforge.net/projects/libdbi-drivers>) package. Please follow the Cygwin-specific instructions

that accompany both packages to build and install the software. Prebuilt binaries for both libdbi and libdbi-drivers are available here (<http://libdbi.sourceforge.net/downloads/>).

- If you need the BibTeX import filter, you'll also want to have the btparse (<http://www.ctan.org/tex-archive/biblio/bibtex/utils/btOOL/>) library. This builds out of the tarball on Cygwin.

4.5.2. Installation from the sources

Note: Depending on the permission settings, you may have to run the installation steps proper (as opposed to the build steps) as an administrator.

1. Unpack the sources in a convenient directory: **tar -xzf refdb-x.y.z.tar.gz** (the exact filename depends on the version).
2. If you do not want to build in the source directory, create an empty build directory.
3. Change into the new `refdb-x.y.z` source directory or into your separate build directory.
4. **./configure --help**

This command will display a list of things you can customize. If you build in a separate directory, use the relative path to `configure` in the source directory. Some important options are:

Note: All paths and URLs in the following options should be entered without a trailing slash.

--prefix=PREFIX

By default, all files will be installed in the `/usr/local` tree. Use this option to use a different install root, e.g. `/usr` or `/opt`.

--datadir=DIR

The data files will be installed in `/usr/local/share/refdb` unless you use this option. The data will be installed in the directory `DIR/refdb`. That is, specifying `"--datadir=/usr/local/share"` is equivalent to the default behaviour. The configuration variable `refdblib` (which will be automatically generated in the example configuration files) must point to the `refdb` data directory.

--sysconfdir=DIR

The global configuration files will be installed in `/usr/local/etc/refdb` unless you specify a different directory here.

--with-libdbi-lib=DIR

Use this option to specify the directory that contains the libdbi library if it is not in the default library path.

--with-expat-lib=DIR

Use this option to specify the directory that contains the expat library if it is not in the default library path.

--with-btparse-lib=DIR

Use this option to specify the directory that contains the btparse library if it is not in the default library path.

--with-sgml-declaration=PATH

Specify the full path to a suitable SGML declaration for your SGML files. If this option is not used, reftdb will use its own copy of `docbook.dcl` stolen from the DocBook DTD distribution. This SGML declaration also works for a variety of other DTDs.

--with-xml-declaration=PATH

Specify the full path to `xml.dcl` which is the SGML declaration for XML files. If this option is not used, reftdb will use its own copy of `xml.dcl` which should work just fine.

--with-docbook-xsl=PATH

Specify the full path to the root directory of the DocBook XSL stylesheets.

Note: This option is required only on systems that do not maintain XML catalogs. If your system is set up properly to resolve public identifiers like those in the XSL stylesheets by XML catalogs, leave out this option. **configure** checks whether the required stylesheets are accessible, so watch out for error messages. If the stylesheets can't be found, either install them, fix your catalogs, or use this option to hardcode the path.

--with-tei-xsl=PATH

Specify the full path to the root directory of the TEI XSL stylesheets.

Note: The TEI Consortium (<http://www.tei-c.org>) ships `tei-xsl-5.2.9.zip` (http://sourceforge.net/project/showfiles.php?group_id=106328&package_id=141124) which contains the stylesheets for both p4 and p5. The root directory which you want to specify here is the directory which contains the p4 and p5 subdirectories, e.g.
`/usr/local/share/xsl/tei-xsl-5.2.9.`

--with-classpath-root=PATH

The refdbxml script assumes that all Java classes for the Java parsers and xslt engines are stored in a class repository, i.e. all in the same directory. Specify this directory with this option. If you keep the relevant Java classes in different directories, either create symlinks or customize refdbxml manually.

--with-refdb-url=URL

The HTML files and templates for the web interface require absolute URLs matching your web server setup. Specify the URL of the directory that contains the refdb HTML files. If you follow the Apache setup instructions in this manual, this will be a string like "http://www.mycomp.com/refdb", where "www.mycomp.com" is the hostname of your computer running Apache and "refdb" is the directory alias used for the refdb web pages.

Note: This URL affects the links in the HTML files and templates. A wrong or missing value of this URL will break the navigation as well as the access to stylesheets and javascript files.

--with-var-dir=PATH

Use this option to specify a directory where refdbd can write its PID file (a file containing the process ID). By default, `/var/run` will be used.

--with-log-dir=PATH

Use this option to specify a directory where refdb programs can write log files to, if logging is directed to a custom file. By default, `/var/log` will be used.

--with-db-server=DBS

By default, all configuration files are preconfigured for use with the MySQL database server. Use this option to preconfigure the configuration files for a different database server (currently mysql, pgsql, and sqlite are recognized as values for DBS).

Note: This option does not modify the binaries that you will create. These will work with all supported database engines regardless of what you select here. The only things affected by this option are a few settings in the refdbdrc configuration file. You can change them after the installation anytime.

--with-main-db=dbname

refdb uses one main database to store citation styles and other stuff. There is exactly one such database per installation with the default name `refdb`. The only reason to change this name is if you want to run this and an older, incompatible version of refdb in parallel.

Finally, run `./configure` with any options that you need. If you build in a separate directory, use the relative path to `configure` in the source directory. A complete set of options might look like this if you use PostgreSQL:

```
$~ ./configure --with-refdb-url=http://yourbox.com/refdb --with-db-server=pgsql --with-var-dir=/var
--with-log-dir=/var/log --with-libdbi-dir=/usr/local/lib --with-docbook-xsl=/usr/local/share/sgml/
--with-tei-xsl=/usr/local/share/xsl/tei
```

5. Run **make**. This will create the executables and adapt scripts and other files to your local installation.
6. **make install** will copy the application server and the clients to `/usr/local/bin` and the data files to `/usr/local/share/refdb` unless you chose different directories in the configure step.
7. To finish the installation, please follow the instructions in the section Finishing the refdb installation below.

Now you have everything in place to use the refdb clients and the application server from the command line. To install the application server as a service, the following additional steps are necessary:

1. Install the service with the command **cygrunsrv -I refdbd -p /usr/local/bin/refdbd -a '-s'** (adapt the paths to your local installation if necessary).
2. Start the service with the command **cygrunsrv -S refdbd**. To stop the service, use the command **cygrunsrv -E refdbd**. If you prefer, you can also start and stop the service with NT's own tools: Either use the NT system control panel or use the NT commands **net start refdbd** and **net stop refdbd**.

With this setup, the service will be automatically started at system startup. See **cygrunsrv --help** for additional options.

4.5.3. Installation of prebuilt binaries

If you decide to grab the prebuilt binaries, the installation will be a little bit faster. The binaries are accompanied by a copy of libexpat. Please make sure to read the aptly named `Readme` files as they may have newer or additional information.

1. Unpack the archive in the root directory (`/`) with the command **tar -xzf refdb-cygwin-bin-x.y.z.tar.gz** (the exact archive name depends on the version you use). This will extract the files into the `/usr/local` hierarchy. The binaries and scripts go to `/usr/local/bin`, the Document Type Definitions, stylesheets, styles and example configuration files go to `/usr/local/share/refdb` and its subdirectories.

2. Unpack the prebuilt binaries for both libdbi and libdbi-drivers, available here (<http://libdbi.sourceforge.net/downloads/>), in the very same way.
3. Add `/usr/local/share/refdb/refdb.cat` to your `SGML_CATALOG_FILES` environment variable.
4. Create the configuration files in `/usr/local/etc`. Sample configuration files with a `.example` extension are created in the same directory during the installation. Provide personalized copies for the user's home directories as needed.
5. To install `refdbd` as a service, follow the instructions above.

4.6. Installation on Windows 95/98/ME

Most of the things said for Windows NT/2000/XP are applicable as well for Windows 95/98/ME. The main difference is that the latter operating systems do not support the idea of a daemon or service. The closest thing you can do to running a service is to put a shortcut to `refdbd` into the startup folder. This is not much of a problem if you run the `refdb` clients on the same computer as `refdbd`. However, Windows 95/98/ME are not suitable to handle `refdb` requests from other computers because `refdbd` will not run if nobody is logged in.

4.7. Other operating systems

`refdb` has not been ported to other operating systems yet. As no esoteric features of the C language are used, you may just try to run `./configure [options] && make` and see what happens. Chances are that it either works or can be convinced to work with just a few modifications. If you manage to run `refdb` on a platform not described here, a short notice with the porting instructions would be greatly appreciated.

4.8. Finishing the `refdb` installation

This step is necessary for all platforms. You have to create some databases for `refdb` to start with and you should make the `refdb` SGML/XML support files known to your system.

4.8.1. Configuring your database server

The default installation of all supported database engines should be just fine for running `refdb`. However, in some cases a little extra work is needed.

4.8.1.1. MySQL

The MySQL engine earlier than version 4.1 supports only one character encoding per server instance. The default encoding is ISO-8859-1, aka Latin-1. If you prefer a different encoding, you have to configure the server at startup. Either use the **mysqld** command-line option

`--default-character-set=charset`, or add a "default-character-set=charset" entry to a suitable MySQL configuration file. A list of available encodings is usually installed as

`/usr/local/share/mysql/charsets/Index`. In MySQL versions 4.1 and later each database (in fact, each table) may have one of various character encodings.

For security reasons many default installation allow only local connections. If `refdbd` has to connect to the database server from a different box, make sure to remove the `--skip_networking` option from the MySQL start script or from the appropriate MySQL configuration file.

4.8.1.2. PostgreSQL

Most default installations of this database server allow only local Unix sockets connections due to security concerns. However, the `refdbd` application server will always talk to the database server via a TCP/IP connection. Please make sure to start **postmaster** with the `-i` command line option to switch on TCP/IP support.

4.8.1.3. SQLite

The embedded database engine SQLite supports two character encodings as a compile-time option: ISO-8859-1 (Latin-1) and UTF-8. The former is the default if you don't use any configure options and if you use prebuilt binaries. If you need Unicode support, you'll have to recompile SQLite using the proper configure switch.

4.8.2. Creating the databases

The `refdb` database contains common information that is shared by all reference databases.

4.8.2.1. MySQL

Note: If `mysqld` (the MySQL database server) is installed on a remote box or if the security settings require it, you may have to use the `-h hostname` and/or the `-u username/-p password` options to run the `mysql` client as shown below (most fresh MySQL installations use "root" with no password as the default database administrator). `mysqld` needs to be up and running and you need the appropriate permissions, of course. See the MySQL documentation for further details.

- In a command line window, run the following command to create the database "refdb":

```
~$ mysql -u root -e "CREATE DATABASE refdb"
```

- Then create the tables and fill in the data. For MySQL older than 4.1, run:

```
~$ mysql -u root refdb < /usr/local/share/refdb/sql/refdb.dump
```

For MySQL 4.1 and later, run this instead:

```
~$ mysql -u root refdb < /usr/local/share/refdb/sql/refdb.dump41
```

Adapt the path to the script accordingly if you configured refdb to put the data directory somewhere else.

The above command will create the tables using the MyISAM engine. This is the fastest of the supported engines, but it does not support transactions. If you prefer to use the InnoDB engine instead, use this command:

```
~$ sed 's/MyISAM/InnoDB/' < /usr/local/share/refdb/sql/refdb.dump41 | mysql -u root refdb /usr/local/s
```

See the MySQL documentation (<http://www.mysql.com>) for further information about the table engines available with MySQL.

4.8.2.2. PostgreSQL

Note: If `postmaster` (the PostgreSQL database server) is installed on a remote box or if the security settings require it, you may have to use the `-h hostname` and/or the `-U username` options to run the `psql` client as shown below (most fresh PostgreSQL installations on Unix-style systems use "psql" with no password as the default database administrator. The Cygwin port of PostgreSQL uses the name of whoever installed the package, usually "Administrator". On Debian you need to be logged in as user "postgres": first **su root**, then **su postgres**). `postmaster` needs to be up and running and you need the appropriate permissions, of course. See the PostgreSQL documentation for further details.

- In a command line window, run the command:

```
~$ createdb -U pgsq1 -E UNICODE refdb
```

The data that you will import in the following steps are UTF-8 data. If you wish to use a different encoding, convert the dump file and adapt the above command accordingly.

- Then run this command:

```
~$ psql -U pgsq1 refdb < /usr/local/share/refdb/sql/refdb.dump.pgsq1
```

Adapt the path to the script accordingly if you configured refdb to put the data directory somewhere else. This SQL script will generate the necessary table definitions and fill in a few values. PostgreSQL will notice you that it is going to truncate a few identifier names. It is safe to ignore these messages.

4.8.2.3. SQLite

- The default database directory is `/usr/local/share/refdb/db`. `refdb` will look here unless you selected a different data directory when configuring the application. If you want to keep your databases somewhere else, use the `dbpath` variable in `refdbdrc` and modify the following instructions accordingly.

```
~$ cd /usr/local/share/refdb/db
```

- Run the following command to create the database and load the data:

```
~$ sqlite refdb < /usr/local/share/refdb/sql/refdb.dump.sqlite
```

4.8.3. Configuration files

Now is the time to create the global configuration files described in the configuration file section. Create these files in `/usr/local/etc/refdb` (or whatever you chose during configuration). It is recommended to copy and modify the commented example configuration files in the same directory. These files are installed with the suffix ".example" to avoid overwriting existing configuration files. All required paths are automatically configured during the installation, so these files are a good starting point for your local modifications.

4.8.4. The SGML/XML support files

The `refdb` package comes with a few additional scripts and stylesheets for the creation of bibliographies. These files are installed in the package data directory (usually `/usr/local/share/refdb`) and its subdirectories, but you should spend a little time to integrate them into your SGML system.

To this end, add the catalog file `/usr/local/share/refdb/refdb.cat` to your `SGML_CATALOG_FILES` environment variable. This is a master catalog with `CATALOG` directives for all catalog files supplied by `refdb`.

4.8.5. The shell scripts

The `refdb` shell scripts (`db2ris`, `refdbjade`, `runbib`, `refdbxml`, and the Perl scripts `tex2mail` and `marc2ris.pl`) were installed in `/usr/local/bin` unless you chose a different install root during configure. As `refdb` attempts to insert the correct settings during the build process, it should not be necessary to manually customize these scripts. If you still want to fiddle with the settings, the variables are clearly marked within a “user-customizable section” at the top of each script.

Note: If you want to use a Java XSL processor with the `refdbxml` script, you'll have to check the value of `CLASSPATH` in the script. The value must match the actual location of your `.jar` files and the current versions you've installed.

4.9. Testing your installation

Now that you got this far, you surely want to see whether your setup actually works. To this end, you may want to run the following tests and see what happens. For your convenience you should perform this test in a graphical environment running at least two console windows.

1. For our first tests it is desirable to have debug information directly available. Therefore you should start `refdbd` from the root account with some special parameters. Before going ahead you should make sure that you do not have a `refdbd` process already running, e.g. because you've setup your system to start that daemon at system startup. The following command will kill any `refdbd` processes. It will do no harm if no such process is running.

```
#~ refdbctl stop
```

If you see an error message saying “`refdbctl: command not found`” then your installation failed or your `PATH` environment variable is not set properly.

2. Now that we know that there is no other `refdbd` process around, we'll start a fresh one with special properties: We'll run it in a separate xterm as a standalone application and have it write debug information to `stderr`. This way, `refdbd` will be slow, but we directly see what's going on.

```
#~ refdbd -s -e 0 -l 7
```

Note: If you try to access `refdbd` from a different box, you should also use the `-I` switch to allow remote connections.

`refdbd` should print some diagnostic information about its startup process and then wait for clients to connect. You may experience problems at this point if the access rights do not allow `refdbd` to create its PID file. This is why we run the test process as root, but if you plan to set up `refdbd` for use with a different account, this is a good time to check and try out these access rights. You may use the `-P` option to provide the path to a PID file that you have write access to in order to run `refdbd` from an unprivileged account.

`refdbd` checks the availability of `libdbi` database drivers during the startup procedure. The following messages indicate a problem with `libdbi`:

Unable to initialize libdbi! Make sure you specified a valid driver directory

You should see a few lines before this message which `libdbi` driver directory `refdbd` attempted to use. Either the default directory is not where the drivers are, or you specified an incorrect driver directory.

Initialized libdbi, but no drivers were found!

The libdbi library was not able to locate or properly load at least one database driver. Please check the libdbi installation and make sure you've installed at least one database driver.

Remember that on most systems the libdbi-drivers package is distributed separately from the libdbi package which contains only the framework, but no drivers. This error also occurs if you install driver versions which are too old for the libdbi framework.

If redbd successfully loads libdbi, it will list the available database drivers. Make sure the driver for the database engine you use is listed.

3. Now switch to a different xterm and start the redba client. The client can basically run from any user account, but the username and password settings have to be those of a database administrator. We're talking about that username and password that you would provide to the command line client of your database server (**mysql** or **psql** for MySQL and PostgreSQL, respectively) for administrative tasks. The username and password settings are either provided by the corresponding configuration file `~/.refdbarc`, or by using the command line options `-u <dbadmin> -p <dbadmin-password>`. In the first case, that is if you set up your configuration file properly, you can just say:

```
#~ redba
```

In the second case you have to use this instead:

```
#~ redba -u <dbadmin> -w <dbadmin-passwd>
```

In both cases the client should start up and wait for your commands with a friendly prompt.

4. At first you might try and see whether the redbd programs read their configuration files properly. To this end, first run the following redba command:

```
refdba: set
```

This will list the current settings of all configuration variables. Make sure these values are what you want. The most important variables are: username and passwd (you won't see the value of the latter for security reasons, though), as well as the host name or IP address of the box that runs redbd. As we currently run both the server and the client on the same box, please make sure that the value of `serverip` is "127.0.0.1".

5. Now try to send a command to the server. We use a command that does not require database access, but it will tell us whether we can connect to the server properly:

```
refdba: viewstat
```

Several things can happen. If you get several lines of output, telling you about the database server used and about a few redbd variables, you're fine. This output may look like the following:

```
You are served by: redbd 0.9.4
Client IP: 127.0.0.1
Connected via pgsq1 driver (dbd_pgsq1 v0.7.1)
to: PostgreSQL 7.2.1 on i386-portbld-freebsd4.3, compiled by GCC 2.95.3
serverip: localhost
timeout: 180
dbs_port: 5432
logfile: /home/markus/prog/refdb/pseudotest/log/refdbd.log
```

```
logdest: 0
loglevel: 7
remoteadmin: off
pidfile: /home/markus/refdbd.pid
```

If you get the following message instead: “could not establish server connection”, check that refdbd is still running and that the serverip setting in refdba is correct. If you try to access refdbd from a remote box, make sure the server was started with the `-I` option. The error may also result from the fact that you failed to create the common database `refdb` or that you did not use the appropriate database server administrator username and password when starting refdba. Finally, your system may lack the runtime client library of the database server.

The error message “failed to connect to database server” shows that your client can talk to refdbd, but that in turn can’t talk to the database server. This may be due to incorrect username/password settings or incorrect permissions with the database servers that support user authentication. If you use SQLite as the database engine, you may have specified an incorrect database directory.

This is all it takes to test the basic functionality of your setup. Everything beyond this is either site-specific setup or mere usage of the tools. Please peruse the manual, especially the hints about getting your database access rights correct.

Chapter 5. refdb administration

This chapter will explain the usage of refdba for the administration of refdb. Please refer to the reference chapter about refdba to learn about command-line switches and interactive commands of this application.

5.1. Create a database

Create new databases with the refdba command **createdb foo**. The new database will be empty initially. If you start adding datasets, the first dataset will have the ID 1.

In order to access the new database, users need to have their access rights set accordingly.

Due to restrictions of the database servers and of the way database names are encoded in refdb SGML/XML documents the names of the databases should be treated as case-insensitive.

Some database engines support different character encodings. In this case you can use the **-E** option to select an appropriate encoding. See the documentation of your database engine for available encodings.

Tip: Your SQL database server may manage more databases than your refdb databases. In this case it may be prudent to use a common prefix for all refdb reference databases that you create. By providing a simple regular expression to the **listdb** command, the execution time may drop considerably.

5.2. Delete a database

Delete existing databases with the refdba command **deletedb foo**.

Warning

The contents of the database as well as the database structure will be gone completely. This is different from deleting all datasets. The latter will leave the database structure intact so you can add references again. The former will force you to create a new database before you can add references again.

Usually this command also removes any permissions to the given database that you have granted to users. If you re-create a database with the same name later, you will have to grant permissions again.

5.3. Add or remove a user

5.3.1. Add users

Adding a user means to allow a user, identified by a database user name and optionally by a password, to read and to modify a particular refdb database.

Note: Both the database user name and the database password may be different from the login name and login password used to get access to the operating system.

Some database engines like SQLite do not support access control. There is no need to add or remove users with these database engines.

refdb uses the access control facilities of the database server. Therefore, the details of adding a user differ slightly between MySQL and PostgreSQL. In any case, refdb is designed to simplify user management as far as possible. This is mainly achieved by combining the information necessary to authenticate a user and the information necessary to grant access to specific database objects into a single command, the **adduser** command of refdba. This command also silently takes care of the fact that users need access to an internal refdb database in addition to the specified reference database.

Database servers use the username, optionally a password, and the host a user tries to connect from to authenticate a user. From the database server's point of view, refdbd is the database client that actually connects to the server, not the refdb clients refdba, refdbc, and refdbib. Database servers usually distinguish between local connections (i.e. by users logged into the same computer) and remote connections (i.e. all connections via TCP/IP). If both refdbd and your database server run on the same computer, all connections will be local and the host-based access control is fairly simple, as both MySQL and PostgreSQL allow local connections by default. All you need to do is:

```
refdba: adduser -d foo jack -N newpass
```

This will allow the new user "jack" to connect to the database server and to access the contents of the database "foo". "jack" has to identify himself by providing the password "newpass" when starting a refdb client.

Note: If a user is already known to the database server, e.g. if he already has access to a different database, you don't have to specify the password again. If you *do* provide a password, the password of that particular user will be changed to the new one.

Things get a tad more complicated if refdbd and your database server do not run on the same computer. As this is where MySQL and PostgreSQL differ, we'll look at these two cases separately.

5.3.1.1. MySQL

On many operating system distributions, MySQL is configured to accept only local connections. Either the `mysqld` process is started with the `--skip-networking` option, or the `my.conf` configuration file contains the corresponding option `skip-networking`. In order to allow remote connections, please remove these options from your system.

MySQL allows to alter the host-based component of access control through the SQL interface. The **adduser** command has an additional option `-H` to specify the host or the network where `refdbd` runs:

```
refdba: adduser -d foo -H % jack -N newpass
```

In this example, `refdbd` may run on any host. You can as well specify a subnet (`-H 192.168.1.%`) or one specific host (`-H mono.mycomp.com`).

5.3.1.2. PostgreSQL

By default, PostgreSQL will accept only local connections. To allow remote connections, the `postmaster` process must be started with the `-i` option. Change your start scripts accordingly.

The host-based component of access control is not available through the SQL interface in this database server. The `-H` option of the **adduser** command is therefore ignored. Instead the database administrator has to edit the configuration file `/home/pgsql/data/pg_hba.conf`. The following entries would:

- Allow local access (i.e. `refdbd` and `postmaster` run on the same computer) to the databases `refs` and `refdb`. The latter is an internal `refdb` database that users must be able to access. Users must provide a password.
- Allow access to the databases `refs` and `refdb` through `refdbd` instances that run somewhere in the network "192.168.1.0". Users must provide a password.

```
#  host  DBNAME  IP_ADDRESS  ADDRESS_MASK  AUTH_TYPE  [AUTH_ARGUMENT]
#  local refdb
#  local refs
#
#  host  refdb  192.168.1.0  255.255.255.0  crypt
#  host  refs   192.168.1.0  255.255.255.0  crypt
```

The default `pg_hba.conf` file allows local access to all databases without password protection. This may not be what you want. The file contains a bunch of helpful comments, though. You may also want to peruse the PostgreSQL documentation for more information about host-based access control.

5.3.2. Remove users

This is again done with the `adduser`, using the `-r` switch. The access rights will be revoked for the specified username and database. No other access rights will be modified. The following command will revoke the access rights of user "jack" on the database "foo".

```
refdba: adduser -d foo -r jack
```

Keep in mind that user "jack" still can access the refdb main database `refdb` and any other databases he was granted access to.

If you use MySQL as your database server and `refdbd` runs on a different box than `mysqld`, you also have to specify the host or network with the `-H` option, just like when you added the user in the first place. See the Add users section for the specifics.

5.4. Configure the application server

There are several ways to configure the application server `refdbd` to your needs. Two of these ways will modify the behaviour at startup, while the remaining two ways will reconfigure the server only for the duration of the current run.

5.4.1. Configuration at startup

There are two ways to permanently change the `refdbd` configuration. First, you can edit the configuration file which holds the default values. Second, you can modify the command line parameters in the script that starts `refdbd` in the first place (e.g. an `init.d` script).

5.4.2. Reconfiguring a running `refdbd` process

Again there are two ways to reconfigure an already running `refdbd` server. The recommended way is to send a HANGUP signal to the server (**kill -HUP *pid***) after editing the configuration file. You can send the signal either manually or use the `refdbctl` script instead. This of course requires that you have a shell on the machine running `refdbd`. This should not be a problem if you sit in front of that machine. If you don't, and you can't do a remote login (e.g. because it is not supported, as on Windows), this won't help. In that case you may use the second possibility. `refdb` supports remote administration via `refdba` if `refdbd` is started with the appropriate parameter. This allows you to use the `refdba confserv` command to reconfigure the server.

Note: The current implementation of the access control to use the `confserv` command requires the `refdb` administrator to be a database server administrator. This is different from being root on the machine.

You can change the settings for the timeout and the settings for the logging facility.

```
refdba: confserv timeout 60
```

This will set the timeout to 60 s, i.e. a client-server connection has to be stalled for 60 seconds before refdbd gives up.

```
refdba: confserv loglevel 7
```

This will set the log level to 7.

```
refdba: confserv logdest 1
```

This will set syslog as the log info destination.

```
refdba: confserv logfile /var/log/mylog
```

This will use /var/log/mylog as logfile if logdest is set appropriately.

5.5. Add, edit, or delete bibliography styles

This task is currently restricted to the system administrators in order to prevent users overwriting each others bibliography styles. A more appropriate solution will be implemented later to allow users uploading their own styles.

Use the refdb command **addstyle** to add one or more bibliography style files to the database. Existing styles with the same name will be overwritten without a warning, so you should check with the liststyle command before you go ahead.

Note: The name of a bibliography style is encoded in the XML file. The filename of the bibliography style is irrelevant here. Use **grep STYLENAME** to see which bibliography styles are defined in a file and check with the liststyle command whether they're already defined in the database.

Warning

refdb uses a non-validating parser to load bibliography styles into the database. It is your responsibility to validate the files before you add them. Adding invalid styles may have undesired and unpredictable results.

The refdb command **getstyle** retrieves one or more bibliography styles as XML documents. In order to edit an existing bibliography style, simply retrieve the style with the **getstyle** command, edit the XML document as necessary, and add it again with the **addstyle** command. As stated above, this will overwrite the existing style definition.

Use the refdba command **deletestyle** to remove one or more bibliography styles from the database. The argument for this command is a Unix regular expression. All styles whose name match the regular expression will be deleted.

5.6. Logging data

The refdb programs can spill out quite a lot of log messages to keep track of what is happening in your programs. This section explains the basics of setting up your message logging.

5.6.1. What to log

The application server refdbd as well as all command-line clients can generate log messages. Message logging is most important for two purposes:

- Keep track of non-interactive programs. This includes refdbd as well as refdbc, bib2ris, and nmed2ris if they are run as CGI applications. It may include all clients if they are run from scripts.
- Track down bugs or user errors.

A useful approach is to log all messages with a log level (explained below) of 6. This would give you a good overview over the usage of these programs but would not clutter the log files with debug information. Switch to log level 7 only if you suspect a bug or some user error and need the full debug information to understand the problem.

For the interactive use of the clients logging is usually not necessary, so you'd use a log level of -1 to prevent logging altogether or a log level of maybe 3 or 4. In the latter case you'd get log messages only if something goes badly wrong. Again, if you encounter bugs or user errors you may switch on debug messages by using a log level of 7.

5.6.2. Destinations

There are three possible destinations the log messages can be sent to. Select the proper destination with either the `logdest` variable in the configuration file or the `-e` switch on the command line. In both cases the values in the following list are accepted. You may use either the numerical value or the case-insensitive string in brackets, e.g. `-e 1` and `-e syslog` are equivalent.

0 (stderr)

stderr is mostly useful for debugging purposes when refdbd is run as a standalone process and when clients are run interactively. stderr does not make much sense if you run refdbd as a daemon (daemons detach from the console at startup, so all output to stderr is lost).

1 (syslog)

Sending the data to syslog integrates the log data with the rest of your system's log output. The log messages will be sent to the user facility, which usually is configured to write to `/var/log/user.log`. See the `syslog(8)` man page for information how to configure the syslog facility.

Note: If you run `reldb` on Cygwin, the syslog messages are sent to the application message list in the NT message logging system. If you are used to going through your log output with tools like `grep` and `awk`, you may find it more useful to write to a custom log file instead.

2 (file)

You can define the full path of a custom log file with either the `logfile` configuration file parameter or the `-L` command line option. By default, `reldb` applications write their log output to `/var/log/<appname>.log`. Make sure to set the appropriate access rights for these log files.

Note: Some libraries used by `reldb`, e.g. the PostgreSQL client library, send log messages to `stderr`. This is not controlled by the log settings of `reldb`. The same PostgreSQL log messages show up in the PostgreSQL log which is usually sent to `/var/log/pgsql`.

5.6.3. Log levels

You can select how verbose the log output of `reldb` will be. You can do this by either setting the `loglevel` configuration file variable or the `-l` command line option to an appropriate value. You may either use the numerical value or the case-insensitive string as explained in the table below. E.g. `-l 5` and `-l notice` are equivalent. Set the log level to `-1` to disable logging completely. Use a value from 0 through 7 to generate increasingly verbose log output. The definitions of the log levels are taken from the include file `syslog.h`:

Table 5-1. Log level definitions

level	verbose	explanation
0	emerg	system is unusable
1	alert	action must be taken immediately
2	crit	the system is in a critical condition
3	err	there is an error condition

level	verbose	explanation
4	warning	there is a warning condition
5	notice	a normal but significant condition
6	info	a purely informational message
7	debug	messages generated to debug the application

Setting the log level to a given value means that all messages with a priority level up to the given value will be logged. E.g. if you set the log level to 6 (which is a reasonable default value), all messages with a priority from 0 through 6 will be logged, whereas messages with a priority level of 7 will be ignored.

Warning

Use log level 7 with caution. The amount of log messages is considerable and sufficient to slow down the application. You should not use this level in everyday use, only to track down bugs or user errors that you may encounter.

5.6.4. Interpreting the log information

If the messages are sent to stderr, only the message proper will be printed. If the messages are sent to the syslog facility, the default format including the process name and process ID is used.

A line in the custom log file looks like this:

```
6:pid=267:Sun Jul 22 18:39:52 2001:application server started
```

You will easily recognize the following fields:

```
message_priority:process_id:time:message
```

message priority

This is the priority assigned to the message by reftdb. The values are explained above.

process ID

This is the process ID (as seen in the **ps ax** listing) of the reftdb process that generated the message. In the case of reftdb this may either be the parent process (the one that generated the above "application server started" message) or one of the children that are forked off to answer client requests.

time

This is the full time and date information when the message was generated.

message

This is the message text proper.

The message entries are sufficiently structured to allow easy access to the information with the standard Unix tools like `awk` and `grep`.

5.7. Security issues

This section briefly discusses some security-related issues that you might want to think about as an administrator. We'll look at the access control provided by the external database servers MySQL and PostgreSQL.

Note: The embedded database engine SQLite does not provide built-in access control. All you can do is use **chown** and **chmod** to restrict access to the database files. There is no way to restrict access through refdb.

5.7.1. Passwords

refdb tries to support the security features of the SQL database servers as far as possible. This includes the username/password-based access rights scheme (not much surprise here). Since version 0.6.0 the passwords are no longer transmitted as plain text between the clients and the server. This means that it now makes sense to keep the passwords secret. There are several ways to specify the password when starting a refdb client. These ways differ with respect to the security of the passwords and are listed here in the order of *increasing* security:

Specify the password on the command line

The password is stored nowhere on the filesystem and thus pretty secure from this point of view. But the full command line can be viewed with the **ps** command by any user on the system, so the unencrypted password is basically world-readable at least for a very brief period until the applications have a chance to hide the string.

Store the password in the personal configuration file

This way the password is protected from other users who habitually run the **ps** command just for the heck of it. But now it is stored unencrypted on the hard drive, and you must make sure that no one else can read the configuration file (no group or world read access).

Specify the password interactively

This is the default behaviour if the password is not specified either in the configuration file or on the command line. The refdb client will ask for the password. This is certainly the most secure way to provide a password, but this won't work if you run the clients unattended via scripts.

5.7.2. Database server access control

It is beyond the scope of this manual to reiterate the security models of the database servers, but you need to keep in mind a few aspects relevant to reldb.

- One component of the database server access control is based on the host from which you connect to the database server. This is partially circumvented by the reldb three-tier design. Keep in mind that only reldb communicates with the database server. Therefore only the host where reldb runs is relevant for the access control. There is currently no system in place for checking whether a client is allowed to connect to the reldb application server from a particular host.
- Both MySQL and PostgreSQL distinguish between local and remote connections. Access needs to be granted separately if you want to use both local and remote connections

Note: On many operating systems the default installations of MySQL and PostgreSQL do not allow remote connections for security reasons. You need to manually allow remote connections as described below.

- The reldb commands to remotely reconfigure a running application server are currently protected by a simple table access test. In any serious database server installation, only the database administrators have read access to certain system tables. The current implementation of this check requires that you have access to this table if you want to run the **confserv** commands. You should be aware that if the access rights are set up improperly, you may also allow everyone and their grandma to stop or reconfigure the reldb application server. If you cannot restrict read access to system tables for whatever reason, you should not enable reldb remote administration (default is off) and use the **kill** command or the reldbctl script instead.
- Most of the reldb commands require database administrator rights.

5.7.2.1. MySQL

If you run MySQL as your database server, these things apply as well:

- When adding users with the **adduser** command the **-H** option has to specify the box where reldb runs as the host, not the box from where the user will run the clients. If you do not specify a host, "localhost" is assumed. This works on some platforms (e.g. FreeBSD 5.4), but not on others (e.g. Fedora Core 4). In the latter case you may have to specify the hostname explicitly.
- To allow remote connections to the database server, the MySQL configuration file `my.cnf` must not contain the option "skip-networking", and the start script must not use the command-line option `--skip-networking`. Many operating systems use one of these methods in default installations to restrict access to local users for security reasons.
- If you install and run MySQL yourself, you should not use the default database administrator account. The name (root) is widely known and by default this account is not password-protected. To make your

database server secure, please create a new database administrator account with a different name and specify a password.

5.7.2.2. PostgreSQL

If you run PostgreSQL as your database server, these hints are for you:

- The host-based part of the PostgreSQL access control is not accessible through the SQL interface. Therefore, the refdba command **adduser** cannot provide the host information (in other words, the `-H` is ignored).

Instead, the host-based part of the access control is specified in the PostgreSQL configuration file, usually `/home/pgsql/data/pg_hba.conf`. On most operating systems, the default configuration allows unrestricted access for all users on the local system, but no remote access. Use something like the following entries to enforce usage of passwords for both local access and remote access from the network 192.168.1.0 to refdb and a reference database "refs":

```
# host DBNAME IP_ADDRESS ADDRESS_MASK AUTH_TYPE [AUTH_ARGUMENT]
local refdb crypt
local refs crypt

host refdb 192.168.1.0 255.255.255.0 crypt
host refs 192.168.1.0 255.255.255.0 crypt
```

As mentioned previously, only the host where refdbd runs is relevant for the host or network entries in the configuration file.

Make sure to read the PostgreSQL documentation to get your access control right.

Note: Please keep in mind that `postmaster` (the PostgreSQL database server parent process) needs to be started with the `-i` option to accept remote connections at all. Most default installations do not use this switch to increase security.

- If you install and run PostgreSQL yourself, you should not use the default database administrator account. The name (pgsql) is widely known and by default this account is not password-protected. To make your database server secure, please create a new database administrator account with a different name and specify a password.

5.8. How to run several refdb instances

In all but a few cases, having one refdbd daemon per network is absolutely sufficient. However, there are a few cases where you may end up running more than one instance of refdbd on the same computer at the same time:

- You want to provide access to more than one database engine. The only good reason to do this may be to test-drive a different database engine.
- You want to test-drive a new refdb release without interrupting the access to your installed version
- You do not want to run refdbd as a daemon serving all users. Instead, each user should be able to run his own copy of refdbd

You'll have to configure each refdbd process individually. If each user starts his own copy of the application server, private `.refdbdrc` configuration files in the users' `$HOME` directory allow a simple setup. If you want to run more than one daemon as non-user-processes, you cannot safely use the **refdbctl** script to control the instances. Instead you should start and stop the daemons manually and use the appropriate command-line options to configure each process individually. The options which need to be set differently whenever more than one refdbd instance runs on the same box are:

- `-D`, `-i`, and `-b` to set the database engine, IP address, and the port it listens on, respectively. Only required if you fiddle with two different database engines.
- `-e` and `-L` to use separate log files for the two instances. This may be easier to evaluate than using `syslog`, although the processes can be distinguished by means of their process ID.
- `-P` to use separate PID files
- `-p` to set the port each refdbd instance listens on. The ports must be different, otherwise clients can't select which instance to connect to.

On the client side you'll need only set the `port` configuration variable or use the `-p` command line option to select the application server instance that you want to connect to.

5.9. Backup your data

If your database crashes for whatever reasons, the first question you'll hear is: "You've got a backup, don't you?". At this point in your life you shouldn't even think about *not* having a backup.

There's basically two ways to create a backup of your reference data: Use a `risx` dump or use a `SQL` dump. The `risx` dump is slower but it would allow you to transfer the data to a refdb installation using a different database server. The `SQL` dump is faster but you're tied to the database server you're using. If you use `SQLite` as your database backend, you may simply grab copies of the database files. Each database is stored in one file. This file holds all required information and is platform-independent, so it is suitable for backup purposes.

Warning

Backups are useful only if you create them regularly. You should work out a schedule based on how often the contents of the database change. Think about setting up a **cron** job for nightly or weekly snapshots.

5.9.1. Creating a risx dump

All you need to do is to retrieve all references and write them to a file in risx format. Use the **getref** command in refdbc for this purpose:

```
refdbc: getref -t risx -s NOHOLES -o all.ris :ID:>0
```

The `-s NOHOLES` option will dump skeleton references for references that you deleted from the database. This simplifies re-creating the same ID values in a new database.

5.9.2. Creating a SQL dump

This is best done with the command line tools shipped with your database server. You should consult the manual of your database server, but the following commands should be all it takes:

If you run MySQL as your database server, run:

```
$ mysqldump -u root -p --opt dbname > dbname.dump
```

This assumes that "root" is the name of your database administrator account and "dbname" is the name of your database.

If you run PostgreSQL instead, use this command:

```
$ pg_dump -u pgsql -C dbname > dbname.dump
```

This assumes that "pgsql" is the name of your database administrator account and "dbname" is the name of your database.

To restore a database from a dump, use the command line clients shipped with your database server. For MySQL, the required sequence is:

```
$ mysql -u root -p -e "CREATE DATABASE dbname"
```

```
$ mysql -u root -p dbname < dbname.dump
```

whereas the following would do the trick with PostgreSQL:

```
$ psql -u pgsql template1 < dbname.dump
```

Tip: SQL dumps are also well suited to create backups of refdb, the common refdb database.

III. User manual

Chapter 6. Overview of the refdb clients

There are several client applications, each one performing a self-contained subset of tasks. This chapter provides a short overview over common features of the clients. Please refer to the reference chapters about refdb, refdbc, and refdbib for the nitty gritty details.

Note: This manual covers only the clients shipped with refdb. Please visit the refdb website (<http://refdb.sourceforge.net>) for other options. Currently there is a PHP-based web interface and tools for both Emacs and Vim which turn these editors into integrated authoring tools with bibliography support.

The clients serve the following purposes:

refdba

The system administrator's tool to create or delete databases, set access rights, and view all kinds of statistics.

refdbc

The client for adding, editing, deleting, and searching reference entries.

refdbib

The tool to create bibliographies.

refdbc and refdb can be run in an interactive mode where they provide their own command line, similar to a command-line FTP client for example. They use the GNU readline library with its powerful and convenient editing and history capabilities. In a nutshell, you can edit commands on this built-in command line as you can do in your bash or in the Emacs minibuffer. You can scroll through the history of your previous commands with the **arrow** keys, you can use the **tab** key to complete command names and filenames, and you can use **Ctrl-r** to search the history with a regular expression. For further information you should consult the readline manual.

refdba and refdbc can also be run in batch mode. You can specify one of the built-in commands as an argument to the **-C** option and pipe or redirect data into stdin. This is very useful if you want to automatize tasks using shell scripts.

refdbib does not have an interactive mode and can be run only in batch mode. The result will be written to stdout. However, you'll hardly ever run refdbib directly, as refdb ships with convenient scripts and Makefiles for this purpose.

6.1. Quickstart guide

These are the essential steps to get up and running as a refdb user:

1. Ask your refdb administrator to grant access to an existing reference database or have him create one for you.
2. If the application server reftdbd is not installed to run as a daemon, start it now: **reftdbd -s &** (see Starting reftdbd for more options).
3. Start the reftdbc command line client to manage and retrieve references.
4. Add references to the database.
5. Retrieve references according to your search criteria.
6. Create bibliographies with the references in your database.

Note: There is a tutorial for first-time users available at the refdb documentation page (<http://refdb.sourceforge.net/doc.html>).

6.2. The interactive mode of reftdba and reftdbc

For informations how to start these clients from the command line, please see the sections about reftdba and reftdbc.

If you did not specify a password in the corresponding configuration file or with the `-w` command line switch (see later in this chapter), the clients will at first prompt you for a password. This is the password to access the database which may be different from your login password. To protect your password from prying eyes, it will not be echoed on the screen while you type it, so no one can even see the length of your password. If your database account was set up without password protection, just press enter now.

Note: If you run a client in batch mode and send data to stdin, the interactive password prompt will not work. Please use the configuration file or the `-w` option to specify the password in this case.

Then you will be faced with a command prompt similar to the one in your shell. The client waits for you to enter commands. If it just escapes you which commands are available, use the **help** command (or **?** for the lazy among us) to see a list.

The general format of a command is:

command [-x *option-argument...*] [*argument...*]

Commands may thus have options and arguments. The options have the general style `-x` [*option-argument*]. They consist of a dash followed by one character. Options are case-sensitive. The option may be followed by an option-argument. This has to be enclosed in quotation marks if it contains spaces. The order of the options and arguments on the command line is arbitrary. Options without arguments can be combined, e.g. you may write `-xy` instead of `-x -y`.

The command line of the refdb interactive clients uses the GNU readline library. This offers you a functionality similar to the bash command line or the Emacs minibuffer. By default the Emacs keybindings are active, allowing you to conveniently edit your command. You can scroll through a history of previous commands. You can use the **Tab** key to try a completion of a partially entered token. If it is the first token on the command line, the clients try to complete a command. If it is any other token, the clients try to complete a filename.

When a command is sent to the application server, the client waits for the result of the command. This may take some time, depending on the load of the application server and of the database server. Network problems or a server crash may result in a timeout. The client will in this case return to the command prompt after a configurable time.

If everything runs smooth, you will see the results of your command. Short results of only a few lines will simply be sent to stdout. Results that may cause the terminal to scroll parts of the result off the screen will be piped through a pager of your choice. Depending on the pager, you can jump back and forth in the results and perhaps save the result or part of the result in a file or pipe it to another program. It may be necessary to press some button (like **q** in less) to get back to the command prompt when you're done. Instead of a pager you can specify any other program that accepts data on stdin, so you can e.g. preprocess the output with **grep** before you display it with a pager ("plumbing", i.e. piping through several applications, is of course possible). Some commands have a switch to redirect the output into a file without displaying it on the screen.

If you output large amounts of data through a pager, the pager may refuse to read on before you scroll through. In this case, a timeout would result and the data would be incomplete. The best way to deal with large amounts of data is to send them to a file and inspect them afterwards. You can use the command **tail -f outfile** to view the data in another console window while they arrive.

6.3. The non-interactive mode of refdb and refdbc

These are the main features of the non-interactive (or batch) mode:

- You have to specify a command name with the **-C** option. Specify all options that the command would take in an interactive session as additional options on the command line. The client will execute this command and exit.
- The clients accept the input for some commands on stdin. If input is available on stdin, be it through a pipe or from a redirected file, this input will be used and any input from a file with the **-f** option will be ignored. As mentioned previously, the automatic detection of data on stdin does not work on a few platforms like Windows/Cygwin. In this case, please use the **-f stdin** option to make the client read from stdin.
- The main purpose of the batch mode is to run the clients from shell scripts (or Perl or PHP if you prefer). You can figure out scripts of arbitrary complexity to interact with your reference database.

Note: You will still be prompted for a password unless you specify one with the **-w** or in the configuration file. Both ways are considered less secure than typing it on the password prompt, but

this is a tradeoff if you want unattended operation. Please keep also in mind that the password prompt will fail if you send data to stdin.

6.3.1. Example

The following command runs refdbc and uses the **getref** command to retrieve references by the author "Miller", using the database "refs" and requesting the RIS output format as well as the abstracts.

```
~$ refdbc -d refs -t ris -s AB -C getref :AU:=Miller
```

6.4. The non-interactive mode of redbib

refdbib is a somewhat simpler "filter"-style application. It accepts data either at stdin or from a file whose name is specified on the command line. The output always goes to stdout and can be piped to other applications or redirected into a file as you need it. For an overview of the command-line options, please see the section about starting redbib. Keep in mind that the runbib script offers a simpler and more versatile way to run redbib.

6.5. Common command-line options for all clients

All clients share a common set of command-line options. The command-line options override the corresponding settings in the configuration files. Please see the chapters about the individual applications in the Reference manual for all application-specific options.

All clients use the GNU getopt library to parse the command line. Switches can be combined (`-abc` is the same as `-a -b -c`) and the sequence is arbitrary (`-a -b` is the same as `-b -a`). All options are case sensitive (`-A` is *not* the same as `-a`).

-c *command*

The command line of the pager that is to be used. Instead of a pager you can of course specify any valid command that accepts data on stdin. Use "stdout" to request data output to stdout. This is the default, but you may want to specify it on the command line if you need to temporarily override a default pager setting in your configuration file.

-e *logdest*

logdest can have the values 0, 1, or 2, or the equivalent strings "stderr", "syslog", or "file", respectively. This value specifies where the log information goes to. "0" (zero) means the messages

are sent to stderr. They are immediately available on the screen but they may interfere with command output. "1" will send the output to the syslog facility. Keep in mind that syslog must be configured to accept log messages from user programs, see the `syslog(8)` man page for further information. Unix-like systems usually save these messages in `/var/log/user.log`. "2" will send the messages to a custom log file which can be specified with the `-L` option.

-h

Displays help and usage screen, then exits.

-i *IP-address*

Set the IP address of the box which is running the application server (*refdbd*). Instead of the IP address you can also specify the hostname as long as it can be properly resolved by your system.

-l *loglevel*

Specify the priority up to which events are logged. This is either a number between "0" and "7" or one of the strings "emerg", "alert", "crit", "err", "warning", "notice", "info", "debug", respectively (see also Log level definitions). "-1" disables logging completely. A low log level like "0" means that only the most critical messages are logged. A higher log level means that less critical events are logged as well. "7" will include debug messages. The latter can be verbose and abundant, so you want to avoid this log level unless you need to track down problems.

-L *logfile*

Specify the full path to a log file that will receive the log messages. Typically this would be `/var/log/<appname>`.

-p *port*

Set the port of the box which is running the application server

-q

Start without reading the configuration files. The client will use the compile-time defaults for all values that you do not set with command-line switches.

-T *time*

Set the timeout for client/application server dialogue in seconds. A connection with unsuccessful read or write attempts will be considered as dead and taken down after this amount of time has elapsed.

-u *name*

Set the username for the database access.

Note: This username need not be identical to the login name of the user. This is the username required to access the database server.

-v

Prints version and copyright information, then exits.

-V

Switches to verbose mode.

-w *password*

Set the password for the database access.

Note: This password need not be identical to the login password of the user. This is the password required to access the database server.

Warning

If you pass the password as a command-line parameter, it will be visible in the process listing with the **ps** command (at least for a brief period of time, until the client overwrites the password string). To avoid sharing your password with the rest of the world, you should either keep your password in your configuration file or use `-w '*'` instead. This will cause the reldb client to ask you for the password interactively (this is also the default if you don't use the `-w` switch at all and don't have an entry in the configuration file). You must protect the `'*'` with single quotation marks to prevent the shell from expanding it into a list of files in the current working directory.

-y

Specify the directory where the global configuration files are

Note: By default, all reldb applications look for their configuration files in a directory that is specified during the configure step when building the package. That is, you don't need the `-y` option unless you use precompiled binaries in unusual locations, e.g. by relocating a rpm package.

Chapter 7. Data input

This chapter explains the ways how you can generate reference and extended note data suitable for refdb databases. We'll cover manual generation of RIS files and automatic conversion of other bibliographic data with the help of input filters. You'll learn about writing risx documents, a new XML data format designed to simplify funneling XML or SGML bibliographic data into refdb. risx is the preferable format to fully utilize refdb's capabilities. Finally we'll cover writing extended notes that allow to append additional information to objects in your database or to categorize references in a very flexible way.

To actually import the resulting RIS datasets into the refdb database, use the `addref` command of the `refdbc` command line client, as explained in the next chapter.

7.1. Writing RIS datasets

7.1.1. Overview

The RIS format is a tagged file format with the following general rules:

- A file can hold one or more references
- Each reference starts with a newline. This also means that every RIS file starts with an empty line.
- There can be only one tag per line.
- The tag must be at the very beginning of the line.
- The tags consist of two capital letters denoting the type, followed by two spaces, a dash, and another space.
- The first tag of each reference is the Type tag (TY -)
- The last tag of each reference is the End tag (ER -)
- The sequence of all other elements is arbitrary.

Note: There is one exception: In the case of multiple authors, editors, or series editors, the sequence of these authors *is* significant. E.g. the first AU tag will be used as the first author, the last one as the last author.

Therefore a minimal RIS file may look like this:

```
①
TY  - BOOK                                ②
AU  - Hoenicka,M.                        ③
TI  - The refdb manual
```

```
PY  - 2001
ER  -
```

④

- ❶ This is the empty line generated by a linefeed character (0x0A) that precedes every RIS citation, even at the start of a RIS file.
- ❷ This is the mandatory first tag, the type specifier. In this case, we're looking at a BOOK entry.
- ❸ This and the following lines are the contents proper of the citation. All additional tags would go here as well
- ❹ This is the mandatory last tag which closes the citation. Although invisible here, this tag also has a trailing space like all others.

7.1.2. Character encodings

The RIS specification has not built-in means to specify the character encoding of the data. Commercial applications apparently expect the data to be encoded as ISO-8859-1, aka Latin-1. refdb does not have this limitation, you are free to use any encoding available on your platform (except UTF-16 and UTF-32). However, you should be aware that this may cause an interchange issue if you plan to use these data in a commercial reference management program. In any case, as the datasets do not specify their encoding, you have to use `-E` option of the `getref` command if your input data use an encoding different from the default (ISO-8859-1).

7.1.3. RIS tags

The following list shows all available tags and their use.

Note: Please keep in mind that a full tag consists of the letters mentioned below followed by two spaces, a dash, and another space. E.g. the first tag below would be written as `"TY - "` in a RIS file.

TY

This tag specifies the type of the reference and must be the first tag of each RIS dataset, preceeded by a newline.

Format: This can be any of the following strings:

- ABST (abstract reference)
- ADVS (audiovisual material)
- ART (art work)
- BILL (bill/resolution)

- BOOK (whole book reference)
- CASE (case)
- CHAP (book chapter reference)
- COMP (computer program)
- CONF (conference proceeding)
- CTLG (catalog)
- DATA (data file)
- ELEC (electronic citation)
- GEN (generic)
- HEAR (hearing)
- ICOMM (internet communication)
- INPR (in press reference)
- JFULL (journal/periodical - full)
- JOUR (journal/periodical reference)
- MAP (map)
- MGZN (magazine article)
- MPCT (motion picture)
- MUSIC (music score)
- NEWS (newspaper)
- PAMP (pamphlet)
- PAT (patent)
- PCOMM (personal communication)
- RPRT (report)
- SER (serial - book, monograph)
- SLIDE (slide)
- SOUND (sound recording)
- STAT (statute)
- THES (thesis/dissertation)
- UNBILL (unenacted bill/resolution)
- UNPB (unpublished work reference)
- VIDEO (video recording)

ER

This empty tag denotes the end of the reference. It must be the last tag of each RIS dataset.

ID

This tag is used to uniquely identify the reference in the database. The value is either the unique ID that refdb generates when a reference is imported into a database, or a unique citation key. The latter can be supplied by the user. If no citation key is specified when adding a reference, refdb will automatically generate a unique citation key, based on the name of the first author and the publication year. refdb will create a unique ID value for internal use regardless of whether a citation key is provided or not.

Note: ID values are always numerical (e.g. "11"), whereas citation keys are alphanumeric (e.g. "Miller1999").

While you are free to choose any reasonable citation key (as long as it is unique within the database), you should not attempt to create a ID value manually. It is ignored when adding the dataset, but it may overwrite an existing entry if you update a reference. Along the same line, you should leave the ID tag alone if you retrieve a dataset from the database and plan to update it. The citation key in the retrieved data set is essential to match the modified data with the copy in the database.

ID Format: Integer >0.

Citation key Format: A string with up to 255 characters. You should use 7-bit ASCII characters only to avoid character encoding issues. If you want to work with SGML documents, the citation keys should be all uppercase.

TI

This is the title of a publication. For BOOK and UNPB references this is the same as the BT tag.

Format: A string with unlimited length.

T2

This is the secondary title of a publication, e.g. the book title for a CHAP reference.

Format: A string with unlimited length.

T3

This is the tertiary title of a publication, e.g. the series title for a CHAP reference.

Format: A string with unlimited length.

AU

Synonym: A1. This is the name of one author of the reference. If a reference has multiple authors, each author is specified with an AU tag on a separate line. The number of authors per RIS dataset is

not limited. The sequence of the authors in the authorlist will be determined from the sequence as they appear in the RIS dataset.

Format: A string with up to 255 characters in the form:

Lastname[, (F.|First)[(M.|Middle)...][, Suffix]]. First and middle names can either be abbreviated or spelled out. Use periods to separate initials, and spaces to separate spelled-out first or middle names. Lastname can be a corporate name. Some examples for valid entries:

- King,B.B.
- Benberg,Steven C.
- Mellencamp,John Cougar,Jr.
- Van Zandt,Steven

A2

Synonym: ED. This is the name of an editor of the reference, e.g. an editor of the book in which a CHAP reference was published. The same formatting requirements as for AU apply.

A3

This is the name of a series editor of the reference, e.g. an editor of a series of books in one of which a CHAP reference was published. The same formatting requirements as for AU apply.

PY

Synonym: Y1. This is the primary publication date.

Format: A string with the format “YYYY/MM/DD/otherinfo”, where YYYY denotes the four-digit year, MM and DD denote the two-digit month and day, respectively, and otherinfo denotes any other information with up to 255 characters. If any of these parts is not available, it can be left out, but the slashes must be present. E.g. “1999///Christmas edition” is a valid string.

Y2

This is the secondary publication date.

Format: A string with the format “YYYY/MM/DD/otherinfo”, where YYYY denotes the four-digit year, MM and DD denote the two-digit month and day, respectively, and otherinfo denotes any other information with up to 255 characters. If any of these parts is not available, it can be left out, but the slashes must be present. E.g. “1999///Christmas edition” is a valid string.

N1

The notes. This can be any form of additional information, like pointers to corrections or editorials, or just personal notes about the contents of the reference.

Format: A string with unlimited length

N2

Synonym: AB. The abstract of a reference.

Format: A string with unlimited length

KW

A keyword. If a publication has multiple keywords, each goes on a separate line preceeded with this tag. Keywords are crucial to find references in larger databases.

Format: A string with up to 255 characters

RP

The reprint status of a reference. This can be any of the following strings:

- IN FILE
- NOT IN FILE
- ON REQUEST MM/DD/YY

AV

The availability information. This is a hint where you can find an offprint or a file containing the reference.

Format: A string with up to 255 characters. This can either be a plain-text description like "methods folder, second drawer from top in the green cabinet on the yellow hallway", or an URL pointing to a file. In the latter case, this field has to start with the string "PATH:" with no space between this and the path proper. Using this feature requires some thought and is therefore explained in a separate section.

SP

The start page of the reference

Format: A string with up to 255 characters

EP

The end page of the reference

Format: A string with up to 255 characters

JO

The abbreviated name of a journal or periodical.

Format: A string with up to 255 characters. The components should be separated by a single space without a period after abbreviated words. If you use periods, these should not be followed by spaces.

JF

The full name of a journal or periodical.

Format: A string with up to 255 characters

J1

The abbreviated name of a journal or periodical (user abbreviation 1).

Format: A string with up to 255 characters

J2

The abbreviated name of a journal or periodical (user abbreviation 2).

Format: A string with up to 255 characters

VL

The volume of the journal/periodical.

Format: A string with up to 255 characters

IS

The issue of the journal/periodical

Format: A string with up to 255 characters

CY

City of publication of a book.

Format: A string with up to 255 characters

PB

Name of the publisher or the publishing company.

Format: A string with up to 255 characters

SN

The ISBN or ISSN number.

Format: A string with up to 255 characters

AD

The contact address, usually the any combination of postal or email address and the phone or fax number of the corresponding author.

Format: A string of unlimited length

UR

The URL of an online version of the reference.

Format: A string with up to 255 characters

U1 through U5

The user-defined fields 1 through 5. These fields are not intended to be filled with random bits of information. Each database should have a set of rules what information is to be stored in these fields.

A possible use for these fields is some relevance indicator (e.g. “#” means low, “#####” means high relevance).

You may also use one of these fields to create the equivalents of “folders” that some other reference databases praise as the panacea to organize your references. Just assign the same value to one of these fields for all references that belong to the same folder. Retrieve them by specifying this value in addition to your other search criteria.

Format: A string with up to 255 characters

M1 through M3

The miscellaneous fields 1 through 3. These fields are used by various reference types to deal with additional information that doesn’t fit anywhere else. M1 is a good place for type or genre information, whereas M3 is suitable to hold the medium.

Format: A string with up to 255 characters

7.1.4. Examples

The following listing shows a few examples of valid RIS datasets. See also the example RIS file in the `/usr/local/share/refdb/examples` directory.

Note: Long entries like abstracts were divided into several lines using slashes. This is to make it more human-readable for this manual and should not be used in real data.

```
TY  - JOUR
T1  - T-lymphocytes from normal human peritoneum are phenotypically /
different from their counterparts in peripheral blood and CD3- lymphocyte /
subsets contain mRNA for the recombination activating gene RAG-1
A1  - Hartmann,J.
A1  - Maassen,V.
A1  - Rieber,P.
A1  - Fricke,H.
Y1  - 1995///
KW  - Peritoneum
KW  - T cell
KW  - T lymphocyte
KW  - lymphocyte
KW  - immunology
KW  - CD3
KW  - human
KW  - Adult
KW  - blood
RP  - IN FILE
SP  - 2626
EP  - 2631
```

JF - European Journal of Immunology
 JA - Eur.J.Immunol.
 VL - 25
 N2 - These findings are compatible with the hypothesis that the adult /
 human peritoneum provides a microenvironment capable of supporting a /
 thymus-independent differentiation of T lymphocytes.
 ER -

TY - BOOK
 T1 - Porphyrins and metalloporphyrins
 A1 - Smith,K.M.
 Y1 - 1975///
 KW - Porphyrins
 KW - Metalloporphyrins
 KW - Spectrophotometry [methods]
 KW - spectroscopy
 RP - NOT IN FILE
 CY - Amsterdam
 PB - Elsevier Scientific Publishing Company
 ER -

TY - CHAP
 T1 - Physiological studies of the natriuretic peptide family
 A1 - Lewicki,J.A.
 A1 - Protter,A.A.
 Y1 - 1995///
 N1 - Atrial Natriuretic Peptide Cardiac synthesis and secretion of /
 ANP Regulation of ANP Gene Expression Regulation of ANP Release /
 ANP Receptors Biologic Actions of ANP Brain Natriuretic Peptide (BNP) /
 BNP Structure Biosynthesis of BNP Biological Actions of BNP C-Type /
 Natriuretic Peptide (CNP) Biologic Actions of CNP Modulators of /
 Natriuretic Peptide Clearance Effects of Clearance Receptor Blockers /
 Effects of Neutral Endopeptidase Inhibitors Role of the Natriuretic /
 Peptides in Physiology and Disease Hypertension Congestive Heart /
 Failure Supraventricular Tachyarrhythmias Acute Renal Dysfunction
 KW - natriuretic
 KW - ANF
 KW - ANP
 KW - receptors
 KW - BNP
 KW - CNP
 KW - hypertension
 KW - congestive heart failure
 KW - review
 KW - cardiac
 KW - regulation
 KW - gene expression
 KW - expression
 KW - brain
 KW - structure

```

KW - biosynthesis
KW - receptor
KW - inhibitor
KW - physiology
KW - renal
KW - study
KW - Peptides
KW - atrial natriuretic peptide
KW - MODULATOR
KW - secretion
KW - Gene Expression Regulation
RP - IN FILE
SP - 1029
EP - 1053
VL - 2
T2 - Hypertension: Pathophysiology, Diagnosis, and Management
A2 - Laragh, J.H.
A2 - Brenner, B.M.
IS - 61
CY - New York
PB - Raven Press, Ltd.
ER -

```

7.2. The Emacs RIS mode

If your editor of choice should be Emacs or XEmacs, the RIS mode (`ris.el`) shipped with `refdb` will make editing RIS datasets a little bit more comfortable. This major mode is still in its infancy, so it will currently do only a few things for you:

- Font-locking will help you to spot syntax errors in the tags. Especially the end tag (`ER -`) is prone to lack the trailing space if you're not careful. The `ris`-mode displays regular tags in blue and the special type (`TY -`) and end (`ER -`) tags in red. Tags will be displayed in the default foreground color if anything is wrong with them, like invalid specifiers, lowercase specifiers, missing or additional spaces.
- `ris`-mode provides three commands to insert datasets ("references") and individual tags as described shortly. All of these commands have in common that they always start a new line after the current line if the cursor is not at the start of a line. Thus you can run these commands from any position of the current line and still get something that makes sense as a RIS dataset.
- Run the command **insert-set** (`C-c-C-s`) to insert a new skeleton dataset (a "reference"). The function will prompt you to enter the publication type. You can use either the auto-completion feature of the minibuffer to enter a valid type or the history feature to select a previously entered type. The function will create a newline, a type tag with the type you selected, and an end tag.

- You can insert a new tag at the beginning of a line with the command **insert-tag** which is bound to C-c-C-t. Use either the auto-completion feature of the minibuffer to enter a valid tag or the history feature to select a previously entered tag.
- You can insert a new line below the current line with the same tag as the current line with the command **duplicate-tag**. This is bound to M-RET. This command is convenient if you add multiple keywords or authors, each of which have to go on separate tag lines.
- You can move between RIS datasets with the commands **backward-set** (C-x[) and **forward-set** (C-x]). You can narrow the buffer to the current RIS set with the command **narrow-to-set** (C-xns). To widen to the full buffer contents again you'd use C-xnw as usual.

Note: ris-mode does not attempt to *validate* the buffer contents. You can create invalid tags, leave out the essential type or end tags, forget about the newline preceeding each dataset and ris-mode will not complain. However, you can spot most errors by looking at the font colors: If the first and the last line of a dataset are not displayed in red, you have a problem (you might be using a monochrome display but that's not what I mean). If any tag (except TY and ER) is not displayed in blue, you have a problem as well.

To install this mode on your system, follow these simple steps:

1. Either copy `ris.el` into a directory which is in your load-path (`/usr/local/share/emacs/site-lisp` is a common place for such files), or include `/usr/local/share/refdb/site-lisp` in your load-path.
2. If you're a speed fanatic, byte-compile `ris.el` with the Emacs command **M-x byte-compile-file path/to/ris.el**. You won't notice a difference on systems later than a 486, though.
3. Put the following code into your `~/.emacs` or into the site-wide `site-start.el`:

```
;; Turn on syntax coloring
(cond ((fboundp 'global-font-lock-mode)
      ;; Turn on font-lock in all modes that support it
      (global-font-lock-mode t)
      ;; maximum colors
      (setq font-lock-maximum-decoration t)))
;; ris mode
(autoload 'ris-mode "ris" "Major mode for RIS bibliography files." t)
(or (assoc "\\ris$" auto-mode-alist)
    (setq auto-mode-alist (cons '("\\ris$" . ris-mode)
                                auto-mode-alist)))
```

7.3. Input filters

refdb gets all reference data from text files. These text files can be from various sources: you can enter them manually, you download files from search engines in the web, or you have data exported from a

different database application. In almost any case, some kind of filtering has to be applied to those files in order to be valid input for refdb.

The input filters currently shipped with refdb are listed below. Please follow the links to the individual filters for further information.

dos2unix

A simple (maybe *too* simple) shell script to convert text files like RIS documents from DOS-style line endings to Unix-style line endings. Most refdb tools need their input files with Unix-style line endings. This is a valuable tool to import reference databases from Windows reference managers.

med2ris.pl

A tool to convert Pubmed data in both the tagged and the XML format to RIS

nmed2ris

A tool to convert tagged Medline data to RIS. This tool is at the end of its life cycle and will eventually disappear from the distribution. Most of the functionality is available in the med2ris.pl script.

bib2ris

A tool to convert BibTeX data to RIS

db2ris

A tool to convert reference data in DocBook SGML/XML documents to RIS

marc2ris.pl

A tool to convert references in MARC format to RIS.

If you need other formats than those listed above, you'll either have to provide your own input filter or search the web for existing filters that convert your data to one of the supported formats. A decent set of filters is supplied by Chris Putnam's bibutils (<http://www.scripps.edu/~cdputnam/software/bibutils/bibutils.html>) package. Another good resource is e.g. Dana Jacob's (<http://www.ecst.csuchico.edu/~jacobsd/bib/>) pages about bibliography software.

7.4. Writing risx datasets

XML documents using the risx DTD (<http://refdb.sourceforge.net/dtd/risx/risx.dtd>) are an alternative way to add datasets to refdb databases. You can use your favourite SGML/XML editor to edit these datasets. You can also use DSSSL or XSLT scripts to transform bibliographic data available as SGML or XML documents to risx.

This section provides a quick outline of risx datasets. For a description of all available elements and their relationships, please visit the risx documentation (<http://refdb.sourceforge.net/risx/book1.html>).

As usual, start the document with the processing instructions, followed by the document type declaration. Make sure to include the character encoding if it is different from the default (UTF-8). The other encodings supported by refdb are UTF-16, ISO-8859-1, and US-ASCII. The first line might then read:

```
<?xml version="1.0" encoding="utf-8"?>
```

The top-level element of a risk XML document is either `ris` (if the file provides multiple datasets) or `entry`, which corresponds to a single dataset. The `ris` element holds one or more `entry` elements. The `type` attribute specifies the type of the reference. These are the same types as described above for the RIS TY tag. The `id` and `citekey` attributes specify a numeric ID (which will only be used if you update references) and a citation key, respectively. The latter should be all uppercase if you intend to use the references with SGML documents.

Each `entry` element contains up to five subelements, the first three of which provide the bibliographic information proper. `risx` distinguishes three levels of bibliographic information. Each `entry` can specify one or more of these levels:

- The `part` element corresponds to the analytical level of bibliographic information. This element specifies the information of a work that has been published as a part of a publication. This can be a journal article in a periodical, or a chapter in a book. The `part` element usually contains information about the authors and the title of the part, as well as volume, issue, and page information.
- The `publication` element corresponds to the monographic level. This is essentially an individual item that you can find on the shelves of a library, like a book or a journal. The `publication` element contains information about the authors/editors and the title of the items.
- The `set` element corresponds to the series level. This provides information about individual items that have been published as a part of a series, like a book about cats within a series of books about mammals. This element usually adds the information about the title of the series and the series editors.

Please refer to the `risk` documentation (<http://refdb.sourceforge.net/risk/book1.html>) for further details about the structure of these elements.

The `libinfo` element contains the local "housekeeping" information of a particular user. Each dataset may contain `libinfo` elements of one or more users who are specified with the `user` attribute. Use this element to specify the notes, the reprint status, and the availability information.

The `contents` element specifies, you've guessed right, the contents of the reference, i.e. an abstract and an unlimited number of keywords for indexing purposes.

7.5. Writing extended notes

Both the RIS and the `risk` formats allow to keep user-supplied notes of unlimited length with each dataset. This is a great way to keep additional explanatory information along with the hard bibliographic data, but this approach is still somewhat limited.

Extended notes are kept separately from the reference data, but there is a mechanism to link each note to an unlimited number of references, author names, keywords, or periodical names. Possible applications of this feature include:

- Write a note about a topic and link it to all references relevant to this topic
- Keep biographic data or alternative (mis)spellings with author/editor names
- Store the impact factor, the official web page, or your personal access information to the restricted part of that web page along with a journal name
- Explain alternative spellings or synonyms of keywords

Searching for notes is similar to searching for references. Notes may have keywords, keys, and a title attached to them to easily find them. In addition, you can search for notes that link to a particular reference, author, keyword, or periodical. The inverse works as well: you can search for references that are linked to particular notes.

Extended notes are XML documents according to the `xnote DTD` (<http://refdb.sourceforge.net/dtd/xnote/xnote.dtd>). The structure of these documents is simple enough to do without a separate documentation. As usual, start the document with the processing instructions, followed by the document type declaration. Make sure to include the character encoding if it is different from the default (UTF-8). The other encodings supported by `refdb` are UTF-16, ISO-8859-1, and US-ASCII. The first line might then read:

```
<?xml version="1.0" encoding="utf-8"?>
```

If you want to write several extended notes in a file, start with an `xnoteset` element. Each individual extended note is kept in an `xnote` element. This element carries up to four optional attributes:

id

An unique identifier supplied by the database engine. This attribute is ignored if you add a new note, but it is respected if you update an existing note.

citekey

This is a unique short title or tag which identifies a note unambiguously but is more convenient to remember than the `id`. If you do not supply a `citekey`, `refdbd` will create one based on the username and the date.

user

This is the name of the user that owns the note. If you do not supply a name, `refdbd` will use the name of the current user, which is most likely what you need anyway.

date

This is a timestamp (YYYY-MM-DD). `refdbd` will insert the current date if you do not supply this attribute.

An extended note consists of an optional `title`, the contents proper encoded in a `content` element, zero or more `keyword` elements, and zero or more `link` elements. The `title` is a short description of the note. The `keywords` serve the same purpose as in references.

The `content` element contains the note proper. The contents of the `content` element is stored by `refdb` as is. It may be plain text or markup. The element uses the following optional attributes:

type

This description of the content type may be used by processing applications to render the contents properly. Store e.g. the MIME type or the name of a DTD/Schema in this attribute.

xml:lang

This attribute specifies the language of the contents.

The `link` element is used to link the note to one or more references, keywords, author names, or periodicals. The empty element uses two attributes:

type

The type of the link target. This may be one of `reference`, `author`, `keyword`, `journalfull`, `journalabbrev`, `journalcustabbrev1`, `journalcustabbrev2`.

target

This attribute specifies the database object that the note should be linked to. In the case of a reference, use the ID or citation key. In all other cases, use the name of the author, keyword, or periodical, respectively.

An example set of extended notes is installed into `/usr/local/share/refdb/examples`.

7.6. Input data mangling

Input data are not stored as a literal chunk of text by `refdb`. If you import a dataset, and later retrieve it using the same format, the dataset is not guaranteed to be the same character by character. Instead, the data are sliced up, sometimes slightly modified, and sometimes `refdb` creates additional information. This section tries to explain what happens to your data behind the scenes, and why this is good for you.

7.6.1. Information that `refdb` generates for you

In a few cases `refdb` fills in some default values if the datasets do not specify them. This happens in the following cases:

- Each reference and extended note will be assigned a unique numeric identifier. This is mainly used internally, but you can also retrieve references and extended notes by their ID. The ID is always created by the database server, there is no way to enforce specific IDs for your datasets.

- Both references and extended notes require a unique alphanumeric key. With a few limitations this is an arbitrary string consisting of letters (at least one) and optional digits. If you do not specify a citation key, refdb will create one automatically. In the case of references, the publication year is appended to the last name of the first author. If this string is not unique, a sequential suffix starting at "a" through "z", then "aa" and so forth, is tested until a unique string is found. The same algorithm is used for extended note keys, but instead of an author name the user name serves as the base.
- If no reprint status is specified, refdb inserts "NOT IN FILE" as the default value.
- If your extended notes do not specify a date, refdb will use the current date and insert that instead.

7.6.2. Information that refdb mangles

Citation keys are supposed to work as ID values in SGML and XML documents. To avoid any character encoding hassles, only the first 127 characters of the US-ASCII character set are permitted. These characters work in most character encodings. Any characters outside of the permitted range will simply be deleted from a citation key, regardless of whether it is specified by the dataset or generated from the author name.

Periodical names and author/editor names receive some special treatment in order to make them usable for refdb. Both periodical names and person names should be provided in a particular format. However, if you retrieve your data from an electronic source instead of writing them from scratch, the names may not conform to the rules. In order to make best use of these data, refdb attempts to normalize the incoming periodical and person names until they conform to the rules.

There is a good reason for this normalization. Consider a periodical name like "The Journal of Biological Chemistry". Different electronic sources may abbreviate this as one of:

- J.Biol.Chem.
- J. Biol. Chem.
- J Biol Chem

Although a human reader does not have a hard time to guess that we're looking at the same journal in all three cases, a database is too stupid to understand this. If you add the periodical abbreviations as they are, you'll end up having three different journal entries. As a consequence, a query like **getref :JO:=J.Biol.Chem.** will miss two out of three papers published in that journal. This is not a good thing.

7.6.2.1. Periodical names

refdb normalizes abbreviated periodical names like this: First, the name is tokenized. Separators are periods and spaces. If a token has a trailing period, it is assumed to be an abbreviated word and used as such. If a token has no trailing period, the token is compared to an internal list of unabbreviated words (see the listword and addword commands for further information about this list). If a match is found, no period is added. If no match is found, the token is assumed to be an abbreviation of something else and a period is added. Spaces after periods will be removed as one separator is sufficient. If we consider the three versions of the journal name above, all versions would be normalized to the first one.

7.6.2.2. Person names

The names of authors and editors are normalized like this: Everything to the left of the first comma is assumed to be the last name and remains untouched. The next item (separated by either a space, a period, or both) is assumed to be the firstname. If it consists of one capital letter, a period is added and any trailing spaces are removed. If the firstname is spelled out, it is used as such. All following name parts to the left of the second comma, if any, are assumed to be middle names. Each part receives the same treatment as a firstname. Finally, everything to the right of the second comma, if any, is assumed to be a honorific or lineage part and used as such. All spaces following either a period or a comma are removed. A few examples should make this procedure clear:

- "Miller, John S" -> "Miller,John S."
- "Chun, H-K" -> "Chun,H.-K."
- "Delorie, DJ" -> "Delorie,DJ"
- "Doe, J S" -> "Doe,J.S."
- "Random,Jane,Jr." -> "Random,Jane,Jr."

The last example shows that your data will not be modified as long as they stick to the input format.

Chapter 8. Reference management

You can manage your references with the `refdb` command line client. We will demonstrate the interactive use of `refdb`, but feel free to use the non-interactive batch mode instead if you prefer. We'll also use a conservative, least error-prone approach whenever there are several ways to achieve the same goal. E.g. we'll use intermediate files instead of piping so we can verify what we got at every stage. Once you're sure about what you're doing, you can still speed up things.

8.1. Add references

8.1.1. RIS datasets

Adding references boils down to running **addref** with proper input files. The input files have to be valid RIS files. They may contain one or more RIS datasets.

You can set some fields to default values with the `-g defaultfile` option of the **addref**. This will add the tags given in the `defaultfile` to the end of every dataset before they are added to the database. This means that these values override any values given in the datasets. This option may be convenient e.g. to set the reprint status of all datasets that you add in one fell swoop instead of editing them individually.

Note: Please be aware that any numerical values of the ID tag are ignored when you add references with the **addref** command. `refdb` assigns an automatically generated ID to every new reference, so you can't enforce arbitrary IDs. You can however use the ID tag to provide an alphanumeric citation key, such as "Miller1999". The citation key should consist only of letters and numbers within the 7-bit ASCII range and has to be unique in the database. Remember that the citation key has to be all uppercase if you want to create bibliographies for SGML documents. If you do not specify a citation key, `refdb` will generate a unique citation key based on the last name of the first author and the publication year.

If you want to overwrite an existing reference with a given numerical ID or a given citation key, use the **updateref** command instead.

8.1.1.1. Example

Let's assume we just downloaded a bunch of references from PubMed (<http://www.ncbi.nlm.nih.gov>) (the largest database for biomedical publications). We saved them in the "Medline" format, ending up with several small files containing one or more references each. How do we get them into the database?

- First we convert all references in the files (we prudently named them `pm001.txt`, `pm002.txt` and so on) to the RIS format and write the result into an intermediate file, using the `med2ris.pl` input filter:

```
~# med2ris.pl -o foo.ris pm*
```

We could now open `foo.ris` in our favourite text editor and make a few changes. We could enter additional keywords, provide personal notes (e.g. why we currently find the article worth the paper it is printed on, something that may not be apparent a few months later), or specify the reprint status and availability.

- Then we switch to (or start) the `refdbc` command line client and type the following command at the prompt to add the references to the database `bar` (use the full path to `foo.ris` if necessary):

```
refdbc: addref -d bar foo.ris
```

8.1.2. risx datasets

If you prefer the XML data format instead, use the **addref** with the `-t risx` switch. As the data are imported using a non-validating XML parser you should make sure the documents are valid, either by using some nifty feature of your XML editor or by running the document through a suitable tool like **nsgmls**.

Each entry in a `risx` document can specify a numerical ID value and an alphanumeric citation key. The ID is ignored by the import routines. If you want to update or replace an entry by ID, you have to use the **updateref** command instead. The citation key should only use letters and numbers within the 7-bit ASCII range. Remember that the citation key has to be all uppercase if you want to create bibliographies for SGML documents. If you do not specify a citation key, `refdb` will automatically assign a citation key based on the first author and the publication year.

8.1.2.1. Example

You've written a `risx` dataset from scratch and want to import it into your `refdb` database. This is what you need to do:

- First you need to make sure that your document is valid against the `risx` DTD. Unless your XML editor validates your document anyway, you'll have to use an external validator like **onsgmls** which is a part of the OpenSP suite of SGML tools.

```
~# onsgmls -s /usr/local/share/refdb/declarations/xml.dcl foo.xml
```

No news is good news in this case: If your document is valid, the validator will not output anything. If you do get some error messages, go back to your editor and fix the problems.

- Now switch to `refdbc` and run the following command to add the references in `foo.xml` to the reference database `bar`:

```
refdbc: addref -t risx -d bar foo.xml
```

8.2. Find and view references

The `refdbc` command **getref** is your friend to find and display references. The query language that is used to retrieve specific references from the database is explained in detail elsewhere. This section offers a few additional hints.

The general idea of the **getref** command is that you specify what you want and you get back a chunk of text. You can select the format of this chunk and you can specify what you want to do with this chunk. A suitable strategy for most searches works as follows:

- Run your initial query, send the screen output to a pager, and see whether the result is what you need. We use the screen output because it generates the least verbose output and thus is the fastest.
- If necessary, refine your search as discussed shortly until you arrive at the desired list of publications.
- Re-run the previous query, but use the output format and output destination that you need to work with the data. For example, if you want to edit the selected references, you would use the RIS format and write the result to a file with the `-o` option.

`refdbc` has a uniquely simple way to rerun and modify queries. Many reference database allow you to modify previous queries only by adding new restrictions to the end or removing restrictions from the end of a previous query. `refdbc` is more flexible. In the interactive mode, `refdbc` keeps its own history of commands. You can use the **up** key or the **Ctrl-p** combo to recall the previous query. You can press these keys repeatedly to go further back in the history. Use **Ctrl-r** to incrementally search for a previous query by entering a short string. Adding more characters will make the search more specific. Pressing **Ctrl-r** repeatedly after entering some search string will display other matches further back in the history.

In the batch-mode, you can use the history feature of your shell. In either case you can modify, extend, or restrict any previous query as you see fit.

8.3. Delete references

The `deleteref` command accepts one or more reference IDs as its argument to specify the references you want to get rid of. This is quite straightforward if you want to delete one or two references. But what if you have a disjoint list of 17 references, e.g. all publications of an author who was convicted to be fraudulent? A useful strategy to delete such references is to save a query with the **getref** command to a file and use it as a list of ID values for a subsequent run of `deleteref`. This way you can verify by inspecting the file that you will delete the correct references. Use the `-s ID` option to request minimal datasets containing the ID value instead of the citation key in the ID field.

8.3.1. Example

To stick with the example of the fraudulent author, we assume his name is, as in so many examples, John Doe (apologies to all RefDB users with that name).

- First we select all his publications from the database `bar`, request only the ID, and save the result in an intermediate file:

```
refdbc: getref -t ris -s ID -o tobekilled.ris -d bar ":AU:=Doe,J."
```

The result is, once again, a file with plain text that we can check and edit to our needs. E.g. if you want to keep one or two publications in the list, just delete their entries in the intermediate file.

- Then we use the `deleteref` command to actually delete the references associated with our list of IDs:

```
refdbc: deleteref -f tobekilled.ris -d bar
```

8.4. Edit references

`refdb` has no built-in reference editor. The strategy to edit an existing reference is as follows:

- Save the reference(s) that you wish to edit to a file using a command like **`getref -o foo.ris -t ris -d bar ":ID:=1717 OR :AU:=^Miller"`**. This would retrieve the reference with the ID 1717 as well as all references with a Mr. or Mrs. Miller on the author list. If you prefer, use the `risx` output format instead.
- Use your favourite text editor to edit the references in the file `foo.ris`. Make sure you don't touch the *ID* field as this is crucial to match this reference in the next step with the existing copy in the database.
- Use the command **`updateref -d bar foo.ris`** to update the reference(s) in the database.

8.5. Print references

There are two ways to print references:

- Send the output of the **`getref`** command to a printer:

```
refdbc: getref -c lpr :ID:>0
```

You should probably be a bit more selective about the references than shown above unless you want to empty the paper tray of your printer. In any case, this command will print a list of your references in the screen output format. You can use some additional plumbing along these lines to obtain a more beautiful printer output:

```
refdbc: getref :ID:>0 -c 'pr -f | fmt -w 70 | lpr'
```

- The second option is to write the HTML output to a file and use your browser's print capabilities.

8.6. Use the personal reference list

If you share your reference database with other users, the personal reference list is your tool to still have the personalized database that you want. In a way, you can eat your cake and still have it. First of all, you

can limit your database search with the `-P` switch of the `getref` command to those references that you added to your personal reference list. But that's not all. With `refdb`, all fields of a RIS dataset which must be the same for all users (like the title or the journal information), are common and accessible to all users. The information which is likely to differ between users (the reprint status, the availability information, and the notes), are stored separately for each user. If you later retrieve datasets, you will see your own notes for the reference, while your colleague will see his notes. All the "hard" bibliographic information will be the same for both of you, though.

So the personal reference list is essentially a tool to have your own personal information for a dataset and to create a personal, selective view of the database.

Now the question is, how does a reference get into your personal reference list? There are three ways to accomplish this:

1. If you add a new reference to the database with the **addref** command, the reference will be automatically added to your personal reference list.
2. Add an existing reference to your personal reference list with the **pickref** command. This command will not add personal data to the reference, it just serves the purpose to include this reference into your searches if you limit them to your personal reference list.
3. If you use the **updateref** command to modify the information in an existing reference, this reference will also appear in your personal reference list. If you use the `-P` switch with this command, the data of this reference common to all users will remain untouched, but your personal information (reprint status, availability, notes) will be added.

Remember that one user can act on behalf of a different user with the `-U` switch, that commands like **addref** support.

Of course there are also ways to get rid of the references that you once found useful. If you share the database with other users, the first choice should always be the **pickref -r** command. This just removes your personal information and your association with the reference, while it leaves the common information untouched for other users of the database. Only if you know that no one else is interested in a reference, you should think about using the **deleteref** command.

8.7. Global edit references

`refdb` has no special command for a global edit of all references or a group of references. You can get pretty far though with a default file when updating references and with Unix plumbing.

To change existing text:

- Run a search on the references that you wish to change and pipe the output through `sed` into a file, like **getref -t ris -d bar -c "sed 's/foo/faa/g' > refs.ris" "search-string"**.
- Use the command **updateref refs.ris** to update the reference(s) in the database.

To add or change fields:

- Run a search on the references that you wish to change, like **getref -t ris -o refs.ris "search-string"**
- Create a `defaultfile` containing the field(s) that you want to change or add.
- Use the command **updateref -g defaultfile refs.ris** to update the reference(s) in the database.

8.8. Create periodical synonyms

There is no nice function to achieve this yet. The following trick works more or less, though:

- Use **getref** to search for a reference with the journal that you want to add a synonym to. If possible, use the reference that has the most information available for the journal (e.g. full name and custom abbreviation). Use the `-o` option to write the search result to a file.
- Edit the temporary file and add the tags "JO - ", "JA - ", "J1 - ", and "J2 - " as needed. Save the file to disk.
- Use the **updateref** command to change the information in the database, specifying the temporary file as the argument.

8.9. Character encoding issues

The 7-bit ASCII character set originally employed by PC computers in the days of yore turned out to be insufficient for languages other than English. Reference data may require characters not included in the ASCII character set. The string sorting order may also follow different rules. `refdb` supports national character sets as well as Unicode, which is sort of a superset of all national character sets. As a `refdb` user and administrator you'll have to deal with character encoding issues at different levels.

8.9.1. Character encodings of databases

While it is possible to convert the data during import and export (see the following sections), it is still worthwhile to spend a few thoughts about the character encoding used by your reference databases. If possible, use an encoding that ensures a suitable string sorting order for your data. Choosing a proper encoding also avoids unnecessary character encoding conversions when importing or exporting data.

The available encodings are limited by your database engine:

SQLite

SQLite currently supports only ISO-8859-1 (the default) and UTF-8 as a compile-time option. If you install a binary package, it most likely uses ISO-8859-1.

MySQL

This database engine supports a fairly large number of encodings, but versions prior to 4.1 allow only one encoding per server instance. That is, all databases have to use the same character

encoding. Please see the MySQL documentation (<http://www.mysql.org>) for the growing list of supported encodings

PostgreSQL

This database engine supports a variety of encodings as a per-database option. That is, all reference databases may use different encodings. Please see the PostgreSQL documentation (<http://www.postgresql.org>) for a current list of supported encodings.

8.9.2. Character encodings of imported data

We'll have to distinguish two different sorts of data:

RIS

This plain-text format does not have a built-in way to declare the character encoding of the data. Instead you have to use the `-E` option of the `addref` and `updateref` commands to specify the encoding if it is different from the default (ISO-8859-1).

Please note that the import filters `med2ris.pl`, `en2ris.pl`, and to a limited extent also `marc2ris.pl` support on-the-fly character encoding conversion.

risx and xnote

These are XML formats that can use the XML way of declaring the encoding. This is done in the processing instructions, which is the first line in a XML file. Due to a limitation of the parser used for importing XML data, only four encodings are accepted by `refdb`: UTF-8, UTF-16, ISO-8859-1, US-ASCII. If your data use a different encoding, use the `iconv` command line utility (usually a part of the `libiconv` package) to convert your data to one of the accepted encodings.

8.9.3. Character encodings of exported data

By default, data are exported without a character conversion, i.e. the data will use whatever encoding the database uses. If you want the exported data in a different format, request the encoding with the `-E` option. This option is accepted by the `getref` and `getnote` commands of `refdb` as well as by the `refdbib` client. You may request any encoding that your local `libiconv` installation supports. **man 3 iconv** or **man iconv_open** should give a clue which encodings are available.

8.10. Use pdfroot

The `pdfroot` variable allows you to store the paths to PDF or Postscript offprints of your references in a more efficient and more flexible way than a full path.

The AV field of a reference can hold either a plain text description of the physical place where a paper copy can be found (this can be anything that helps you to get hold of the copy, from a room number to a file folder title) or it can hold the path to an electronic version of the document. In the latter case, the string has to begin with “PATH:” and the path itself must follow without a space. This path will be converted into a link in the HTML output, and the neat plan is to open the document the link points to with a simple mouseclick. If you just store the full path of this document, something like

PATH:/home/markus/literature/5503.pdf, you may run into problems fairly soon: If you have to reorganize the folder structure in your directory tree, all paths in the database would be incorrect. If you access the database from a different computer as the one that physically stores your offprints, you will be out of luck with these paths as well.

The idea is to store the invariable part of the path in the reference entry and any variable part of the path in the `pdfroot` variable. The `pdfroot` variable can be set either in your configuration file or with a command line option. Consider e.g. the following situation: You decide for some obscure reason to store the offprints in subdirectories according to the publication year. The full paths will thus be something like /home/markus/literature/1999/2345.pdf and /home/markus/literature/2001/6213.ps. The partial paths in the AV field of the reference entry could then be set to PATH:1999/2345.pdf and PATH:2001/6213.ps, whereas the correct value for the `pdfroot` would be /home/markus/literature/. Simple concatenation by `refdb` will then result in the proper full path.

If you now decide to rearrange your hard drive, all you need to do is to set `pdfroot` properly, as long as you don’t change the directory structure below /literature (if you do *that*, you shoot yourself in the foot anyway).

If you want to access your offprints from a different computer, you have several options:

- You can mount the directory tree that actually holds the data. If the computer that stores your offprints e.g. exports /home as an NFS share, you could mount that to /mnt on your local computer. Then you can set `pdfroot` to /mnt/markus/literature/ to access your offprints.
- You can store the offprints in a location that is accessible with an internet protocol like HTTP or FTP. Assume you keep the offprints in /usr/local/www/data/refdb/offprints/ on a computer that runs Apache (<http://www.apache.org>). Then you could set `pdfroot` to `http://machine.name/refdb/offprints/` and remotely access the offprints through HTTP.

8.11. Interaction with external applications

`refdb` has been designed to be as modular and as lightweight as possible. This means that you need some external applications for various purposes. This chapter briefly discusses how you can simplify your work with these external applications.

8.11.1. Editor

If you want to modify references in your database or write new entries from scratch, you need some suitable editor. Almost any will do as long as it will create the correct line endings: `refdb` wants a plain

Unix-style LF (0x0A), not a DOS-style CR LF (0x0D 0x0A). The prime contenders are vi and Emacs which are available on almost any Unix-style operating system as well as on Windows. For the Windows platform there are also a number of free- and shareware editors which let you select the line endings.

Creating new references is straightforward: Type them in, save them, and use `refdbc` to upload them. If you want to edit existing references in your database, you download them with the `getref` command to a disk file. Open this file with your editor, edit the contents, and save the file again for uploading. If you want to edit several references in a row, you can simply reuse the filename for downloading, then refresh the buffer in your editor (e.g. **C-x C-f RET** in Emacs). After saving the changes, simply recall the last upload command in `refdbc` with the **up** key and press **Enter**.

8.11.2. Viewer

While a pager will do in many cases to view the query results, a web browser is a nice alternative. The `refdb` command `getref` can generate HTML output at your request with the `-t html` option. Save the output to a file and view this file with your favourite web browser. When you run the next query, reuse the filename and hit the `reload` button of your browser to display the new results.

Tip: A web browser is also a convenient way to print references.

Chapter 9. Notes management

Managing notes is very similar to managing references. The commands have analogous names and functionality. If you've familiarized yourself with managing references, the notes part should not be too hard either.

9.1. Add extended notes

Extended notes must be supplied as datasets according to the `xnote` DTD (<http://refdb.sourceforge.net/dtd/xnote/xnote.dtd>). Add them to the database using the **addnote** command. Each file supplied to this command may either contain a single `xnote` element or several of these wrapped into a `xnoteset` element.

Note: The `id` attribute, if any, will be ignored when you add extended notes. The database engine will assign each note a new, unique identifier. The `citekey` attribute will be honored unless there is a clash with an existing note. In this case `refdbd` will refuse to add the note.

If you want to overwrite or update an existing note, use the **updatenote** command instead.

9.1.1. Example

Assume the file `foo.xml` contains a couple of extended notes. The following command will add them to the database "bar" (use the full path to `foo.xml` if necessary):

```
refdbd: addnote -d bar foo.xml
```

9.2. Find and view extended notes

Just as you'd use the **getref** command to locate references, you can use the **getnote** command to view notes. The query language is explained in a separate section.

The notes can be retrieved for screen display, as HTML or XHTML documents, or as `xnote` documents.

Note: Accessibility of notes is affected by the default note sharing setting of `refdbd` and by the properties of the individual notes. See below for further details.

9.3. Delete extended notes

Use the **deletenote** command to remove extended notes from the database. The argument to this command is a space-separated list of ID values.

9.3.1. Example

The following command removes the notes with the IDs 4 and 132 from the database "bar":

```
refdbc: deletenote -d bar 4 132
```

9.4. Edit extended notes

Use the same strategy as you would for references:

- First retrieve the notes you want to edit with the **getnote** command and write the output to a file:

```
refdbc: getnote -d bar -o foo.xml -t xnote :NID:=4 OR :NCK:=biochemistry1999
```

- Now use your favourite XML editor to edit the notes. Keep the `citekey` and `id` elements alone, as at least one of them is required to match the dataset with the existing copy in the database.
- Finally use the **updatenote** command to update the reference in the database:

```
refdbc: updatenote -d bar foo.xml
```

9.5. Link existing notes to other objects in the database

If you want to link an existing extended notes to other objects in your database without modifying the contents of the note, the **addlink** command comes in handy. The command requires at least two arguments. The first argument specifies an existing note by its ID or citation key. The second argument specifies an object in the database, either a reference by ID or citation key, or an author name, a keyword, or a periodical by their names. Removing links works just the same except that you need to provide the `-r` switch.

9.5.1. Example

The following command links the existing note with the ID 5 to two references which are specified by their citation keys:

```
refdbc: addlink -d bar :NID:=5 :CK:=Miller1999 :CK:=Doe2000
```

9.6. To share or not to share extended notes

refdb was written with facilitating the cooperation of users in mind. Therefore extended notes are visible to all users of the system by default. However, situations may arise where you don't want to share any notes or where you want to protect only individual notes from your colleague's prying eyes. refdb supports all these variants. Accessibility of notes is controlled by two filters: the default server setting and the settings in the individual notes.

9.6.1. The refdbd default

The refdbd server can be configured to either treat all extended notes as public or as private by default. Use the `-s` command line option or the `share_default` config file option to set the default mode. See the chapter about refdbd administration for further details. As the names imply, *public* will make all notes available to every user by default. Similarly, *private* will restrict access to the user who added the note by default. Use the server setting to define a default policy for your installation.

Warning

A warning for all admins: switching your server from "private" to "public" without prior announcement *will* alienate your users. Don't even think about it.

9.6.2. The `share` attribute of the extended notes

The `xnote.dtd` (<http://refdb.sourceforge.net/dtd/xnote/xnote.dtd>) defines an optional `share` attribute which offers the same choice of values as the server setting above. If the attribute is set, the extended note will be public or private regardless of the server setting. The server setting only kicks in if a note does not set the `share` attribute. Each user can decide for each note whether or not she wants to share it with the world. The following table summarizes the settings.

Table 9-1. Sharing extended notes

	share attribute "public"	share attribute "private"	share attribute not set
share_default "public"	public	private	public
share_default "private"	public	private	private

9.6.3. Data privacy

The above settings control only the access to the extended notes through the refdb interface. You should be aware that anyone who can access the database directly with an SQL client will be able to read and change the notes of every user, regardless of the settings mentioned above. The simplest way to secure the data is to run refdbd on a server that no user has shell access to. If the database engine access control

is set in a way that users can only connect from the box that runs refdb, they won't be able to connect from a SQL client running on their workstations.

Chapter 10. Bibliographies

The bibliography is the *really* hard part of writing a scientific publication, much harder than generating the data in the first place. This is why refdb tries to help you with this task as much as possible.

Before you go ahead and generate a bibliography you have to teach refdb about what the result should look like. While the default bibliography formatting of e.g. the DocBook or TEI stylesheets work just fine for many documents, it is clearly insufficient for scientific publications: There are literally thousands of possible combinations for the formatting of authors, titles, journal names, page and date informations, and almost each of these possibilities has been adopted by at least one journal or publisher as the one and only citation and bibliography style. The format of the refdb bibliography styles is described in the first section. The next section will then explain how you generate bibliographies and format your documents.

10.1. Quickstart guide

These are the essential steps to publish SGML or XML documents with formatted citations and a formatted bibliography:

1. Load one or more bibliography styles into the common refdb database, using the `addstyle` command.
2. Prepare your source document by declaring an external entity containing the bibliography in the header and by including a reference to this entity in the proper place of your document.
3. Insert citations into your document, preferably using the short notation.
4. Preprocess your document using `refdbxp` (not required if you use the full notation for citations).
5. Use `runbib` to create the bibliography as an external entity.
6. Use either `refdbjade` or `refdbxml` to transform your document to the desired output format.

Note: The last three steps can be run conveniently from a Makefile generated by the `refdbnd` tool. See the description of the `refdbnd`-generated Makefile below.

10.2. Manage bibliography styles

Bibliography styles are defined as XML documents. Each document contains one or more bibliography styles for a particular journal or publisher. The bibliography styles need to be added to the refdb database before they can be applied to your documents. The bibliography style controls, among others, characteristics like:

- Numerical vs. author/year citation style

- Bibliography sorting order (as it appears in the text vs. alphabetical or sorted by ascending or descending publication dates)
- Formatting of author names: First and middle initial with or without periods, with or without spaces, before or after the surname
- Appearance of name, volume, and issue number of journals (bold, underlined, italics)
- Formatting of the bibliographic listing: indentation, font sizes.

Note: The extensive formatting specifications of the refdb bibliography styles are almost wasted on BibTeX bibliographies currently. You still need one of the native BibTeX styles to do the actual formatting until refdb bibliography styles can be exported as BibTeX styles. The current implementation uses only the formatting information of the journal name to allow either the full name or one of the abbreviations to appear in the bibliography. So for the *current* implementation you can get away with just two simple styles that define only the journal name formatting for the “GEN” publication type. These can be found as `bibtex-abbrev.xml` and `bibtex-full.xml` in the `style` directory of the source distribution.

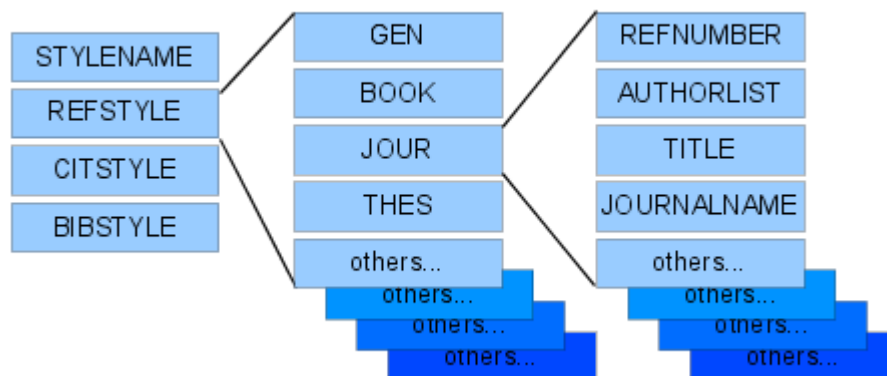
It is admittedly no easy task to write correct bibliography styles from scratch. It may be easier to pick a similar style (if there is one) and modify it to your needs. In this section you will find a brief overview how a bibliography style is put together. For more detailed information, please peruse the separate documentation of the CiteStyle XML DTD (<http://refdb.sourceforge.net/citestylex/index.html>).

10.2.1. Write or modify a bibliography style file

It is recommended to use an XML editor like Emacs/PSGML to write bibliography styles using the CiteStyle XML DTD. This ensures that most XML errors get caught while you write. After you’re done, the text should be validated with a validating SGML/XML parser like `nsgmls` or `onsgmls`.

The `CITESTYLE` element defines a bibliography style for one particular journal or publisher. You can group several styles in one file with the `STYLESET` wrapper element.

Each `CITESTYLE` element contains exactly four top-level elements (Figure 10-1). The `STYLENAME` defines the name of this style. For the sake of simplicity this could be identical with the name of the journal or publisher whose bibliography style it defines, e.g. “J.Biol.Chem.” or “Elsevier”. The `REFSTYLE` element contains the style definitions for the various publication types that can appear in a bibliography, like books, journals, or personal communications. A special case is the type “GEN” which defines a default bibliography style that is applied whenever no specific definition is available for the requested type. Although the DTD does not enforce this, it is strongly recommended to define a “GEN” definition for each bibliography style. The `CITSTYLE` element defines the citation style, i.e. the appearance of the citations in the main text. Finally, the `BIBSTYLE` element defines the properties of the bibliographic listing.

Figure 10-1. Schematic representation of a CITESTYLE element

Each definition for a publication type in turn can contain various elements that define the sequence and rendering of authorlists, publication dates, titles, and so on. This is where the real pain in defining bibliography styles lies. All available elements can hold a `PRECEEDING` and a `FOLLOWING` element which define strings that are inserted before and after the corresponding element, respectively. This can be used to place punctuation characters or brackets wherever such a non-empty element occurs. A special element is `SEPARATOR` which usually also contains punctuation characters. This element is always inserted even if the preceeding or following element is empty.

The `CITSTYLE` element can define three different styles for citations: `INTEXTDEF` for regular citations as well as `AUTHORONLY` and `YEARONLY` for citations that keep the authors in the flow of the text. These elements are equivalent to the definition of a publication type in the `REFSTYLE` element.

Please peruse the separate documentation for the CiteStyle XML DTD (<http://refdb.sourceforge.net/docs.html>) for the details about the individual elements.

Tip: The refdb project also provides a tool to create citation styles interactively. The **refdb-ms** Perl script is available separately from the main project page (<http://sourceforge.net/projects/refdb/>).

10.3. Create bibliographies

This section explains how to prepare documents for use with refdb and how to generate and process the bibliographies. First we'll look at SGML and XML documents, then at LaTeX documents.

10.3.1. SGML and XML documents

DocBook and TEI SGML and XML documents and their refdb bibliographies share many features, so they are treated together in this section. We'll cover how you specify citations, how you generate the bibliography, and how you transform the final document.

10.3.1.1. Prepare the document

The output of the refdbib application will be a `bibliography` element that contains all required references. You can redirect the output into a file and include this file as an external entity at the spot where your bibliography should appear. To achieve this you need two modifications in your document:

1. Extend the document type declaration at the beginning of your document to declare the external entity. The first example is from a DocBook SGML document:

```
<!DOCTYPE BOOK PUBLIC "-//OASIS//DTD DocBook V3.1//EN" [
<!ENTITY bibliography "foo.bib.sgml">
]>
...
```

The second example shows a TEI XML document:

```
<?xml version="1.0"?>
<!DOCTYPE TEI.2 PUBLIC "-//TEI P4//DTD Main Document Type//EN" "http://www.tei-c.org/P4"
<!ENTITY % TEI.general 'INCLUDE'>
<!ENTITY % TEI.names.dates 'INCLUDE'>
<!ENTITY % TEI.linking 'INCLUDE'>
<!ENTITY % TEI.XML 'INCLUDE'>
<!ENTITY bibliography SYSTEM "refdbtest.bib.xml">
]>
...
```

The name of the entity is of course yours to choose, but using “bibliography” as in this example is pretty descriptive.

2. Include the bibliography at the desired spot:

```
...
&bibliography;
...
```

You need to make sure that the included chunk of text is valid at the point where you want to include it. DocBook SGML and XML bibliographies are generated as `bibliography` elements, TEI XML bibliographies are wrapped in `div` elements.

10.3.1.2. Create citations

Creating citations and bibliographies in SGML or XML documents with refdb is very similar to what you would do if you had to manually code the bibliographies - but without the sweat. First you create the citations. Each citation consists of one or more bibliographic references in the text, each of which points to one particular entry in the bibliography. Then you create a bibliography for all cited publications (and possibly some more). For an increased benefit you would certainly also want to create functional links from the citations to the corresponding bibliography entries, which would act as hyperlinks in suitable output formats like HTML or PDF. In real life, you would probably jump back and forth, adding a bibliography entry whenever you add a new citation, and invent suitable ID values for your bibliographic link targets as needed.

Note: The distinction made here between a citation and a bibliographical reference may sound like nitpicking, but it will be important when we deal with citations that contain more than one bibliographical reference.

refdb requires a slightly more formalized approach. You have to stick to a particular syntax when you create the citations, but the good news is that refdb does almost all of the rest. You will usually also create the citations first and let refdb create the bibliography just before you are ready to transform the first draft.

refdb allows two different notations for references:

Short notation

The short notation is, as the name implies, a lot faster to type and thus more convenient, but it requires an additional preprocessing step that adds some small restrictions to the way you write your documents (please see the section about refdbxp for details about these restrictions). The preprocessing of documents using the short notation also automates the issue of first and subsequent citations of a bibliographic entry and it automatically creates the ID values used in multiple citations. Using multiple databases per document is not supported by the short notation currently.

The short notation is fully valid SGML or XML code, without any extensions of the original DTDs. You can use all sorts of SGML or XML processing tools on such documents.

Full notation

The full notation offers full control but requires a lot more typing and thinking. It does not require a preprocessing step before the transformation, though. You need to take care of the issue of first and subsequent citations of a reference, and you have to manually generate ID values for use in multiple citations. You can include references taken from several databases.

Just like the short notation, the full notation is also fully valid SGML or XML code, without any extensions of the original DTDs.

First we'll have a look at the short notation, before we get into the gruesome details of the full notation. Keep in mind that the refdbxp application interconverts the short and the full notation. You can convert

your document back and forth as often as you wish, so you're not limited to the notation that you initially choose. In fact, you can mix both notations in a single document.

10.3.1.2.1. Short notation

In DocBook documents (both SGML and XML), citations are encoded as `citation` elements. To distinguish these from `citation` elements that are not meant to be processed by `refdb`, set the `role` attribute to `REFDB` in all caps. Each `citation` element contains one or more references, separated by semicolons. The trailing semicolon after the last reference is optional, so the following citations are absolutely equivalent:

```
<citation role="REFDB">2;5;9</citation>
<citation role="REFDB">2;5;9;</citation>
```

The list of references can either contain numerical ID values, as in the examples above, or alphanumeric citation keys like in the following example:

```
<citation role="REFDB">miller1999;jones2001</citation>
```

The corresponding syntax for TEI XML documents is quite similar, except that we abuse the general-purpose `seg` element and tag it for use with `refdb` by setting the `type` to `REFDBCITATION` in all caps:

```
<seg type="REFDBCITATION" part="N" TEIform="seg">2;5;9</seg>
```

Again, you can use citation keys instead of the ID values shown in the example above.

The examples shown above will be rendered as "regular" citations. In addition to this you can request author-only or year-only citations. These come in handy if you want to write something like: Jones et al. reported recently (2001)... Both the authors (Jones et al.) and the year (2001) need to be encoded as individual citations as shown in the following example:

```
<para><citation role="REFDB">A:jones2001</citation> reported
recently <citation role="REFDB">Y:jones2001</citation> ...</para>
```

You may have guessed that the prefix "A:" tags a citation as an author-only citation and that the prefix "Y:" means year-only.

Note: These prefixes tag the whole citation, not a particular reference in the citation. Therefore the prefix must be the first thing right after the start tag. Multiple citations using the author-only or year-only style would make no sense anyway.

This is about all you need to know about the short notation. The only thing you must not forget is that you *must* preprocess documents that contain citations in short notation with `refdbxp` before you transform the document to one of the output formats.

10.3.1.2.2. Full notation

The full notation is a lot more complex than the simple notation described above. So unless you have specific reasons to write citations in full notation from scratch, it is more advisable to use the short notation and preprocess your documents with `refdbxp`. The output created by this utility is the full notation described in this section.

The particular syntax of citations and bibliographic references is necessary for two reasons: first we have to tell `refdb` which bibliographic database entry (and probably, from which database) we want to reference. Second, we need to encode which type of citation or reference we want. The exact markup depends on the DTD that your document uses, but the basics are the same.

In both DocBook and TEI documents, these two bits of information are encoded in attributes of elements that create a link from the reference to the bibliographic entry. In order to handle multiple citations correctly, these link elements need to be inside a wrapper element. For a DocBook document, basic citations therefore look like this:

```
<citation role="REFDB">           ❶
<xref linkend="ID1-X">           ❷
</citation>
<citation role="REFDB">
<xref linkend="LITIBP-ID2-X">    ❸
</citation>
```

- ❶ The `citation` element is a wrapper for one or more bibliographic references. The `role` attribute is set to `REFDB` to distinguish this `citation` from other `citation` elements that `refdb` should leave alone. Each `citation` element can contain one or more `xref` elements.
- ❷ Each `xref` element specifies one bibliographic reference. The value of the `linkend` attribute encodes which bibliographic item is referenced (in this case, the database entry with the ID 1) and how the reference should be rendered (see below). It consists of the string "ID" followed by the numerical database entry ID, and a trailing one-letter type specifier ("X" in this case), separated from the rest by a dash. This simple form does not encode the database from which the reference is to be pulled. When generating the bibliography, you will specify a default database from which all references without an explicit database label will be taken from. This form is most convenient if all your bibliographic items are stored in one database.
- ❸ This `xref` element shows the syntax when an explicit database (LITIBP in this case) is specified. The attribute value consists of the database name, a dash, the string "ID", the numerical database entry ID, and the trailing type specifier. This form is mandatory only if you reference bibliographic entries from different databases in the same document (again, one database can be set as the default database in subsequent processing steps, so you could use the simple form for all references to entries in that particular database).

Note: This and the following DocBook examples are given in SGML notation. Keep in mind two things when working with XML documents:

- The empty `xref` elements need a closing slash as in `<xref linkend="ID2-X"/>`.
- All attribute values relevant to `refdb` must be in uppercase. This restriction is imposed by the way citations are currently extracted from the document. It may be dropped in later versions though.

The corresponding syntax in a TEI XML document looks like this:

```
<seg type="REFDBCITATION" part="N" TEIform="seg">           ❶
<ptr targOrder="U" target="ID1-X" TEIform="ptr"/>           ❷
</seg>
<seg type="REFDBCITATION" part="N" TEIform="seg">
<ptr targOrder="U" target="LITIBP-ID2-X" TEIform="ptr"/>❸
</seg>
```

- ❶ The general-purpose `seg` element with the `type` attribute set to `REFDBCITATION` is the citation wrapper for one or more bibliographic references.
- ❷ Each bibliographic reference is specified by a `ptr` element whose `target` attribute encodes the bibliographic entry that is referenced. As explained in the DocBook example, this is the simple form that does not specify the database.
- ❸ This is the corresponding bibliographic reference with the database specified.

Note: You don't have to worry about the attributes in the example which are not mentioned in the explanations. These are TEI default attributes which do not have anything to do with `refdb` (your XML editor will most likely create them automatically for you).

There are several ways to render citations and bibliographic references in the text. You select what you need by a trailing capital letter after the database ID (the "X" in the above examples). `refdb` will create several preformatted strings in the bibliography file which can be linked to by selecting the proper postfix. These preformatted strings have several purposes, as shown in the following table:

Table 10-1. Bibliographic reference types

Postfix	Purpose
X	The most common case. This is the first occurrence of a reference which is to be displayed outside the flow of the text. In numerical citation schemes this will be something like "(2)", in author-year citation schemes this may be rendered as "(Miller et al., 1992)".

Postfix	Purpose
S	This is the same as X, but for a subsequent occurrence of the same reference. This distinction is important for some author-year citation schemes that print the full (or at least a longer) author list at the first occurrence and an abbreviated one at all subsequent occurrences of the same reference.
A	This is the first occurrence of a reference that displays the authorlist inside the flow of the text, like in "Miller et al. reported recently (2001)..."
Q	This is the same as A, but for subsequent occurrences of the same reference.
Y	This type complements the author-only references mentioned above. In numerical citation schemes this is usually rendered like a normal reference, e.g. as "(2)", but in author-year citation schemes usually only the publication date is rendered, as in "(2001)".

Note: The exact formatting of these references, e.g. which citation style is used or which brackets surround the reference, is controlled by the style specification for a particular publication or publisher. This takes effect when you generate the bibliography and transform the final document.

An additional twist comes into play if you have multiple citations, i.e. a citation that contains more than one bibliographic reference. In most cases, all references are displayed inside of one pair of brackets. Some numerical citation styles require that bibliographic references with consecutive numbers be formatted as ranges within the same citation.

Note: Formatting consecutive numbers as ranges kills the links from the reference to the bibliographic item for each reference that make up a range. Any generated hyperlinks will therefore point to one common target for all members of a multiple citation. If this is not desired (e.g. to keep the links alive in a HTML presentation of a scientific document), you may override this behaviour during the transformation of the final document.

In order to format these cases properly, you need to include a dummy element whose sole purpose is to provide a link to an element that contains the combined, preformatted citation string. This is shown for a DocBook document in the following example.

```
<citation role="REFDB">
<xref endterm="IMTHEFIRST" linkend="ID1" role="MULTIXREF">❶
<xref linkend="ID1-X">❷
<xref linkend="ID14-X">
```



```
<xref linkend="ID7-X">
</citation>
```

- ❶ This is the additional `xref` element which is mandatory in multiple citations. The `linkend` specifies the target of a link, which by convention could be the first of the following references. Note that the attribute value does not have a trailing type specifier. The element must have a `role` attribute with the value `MULTIXREF`. You also have to provide an unique value for the `endterm` attribute. This specifies the ID value that will be used in the corresponding element in the `refdb`-generated bibliography that contains the preformatted string for the multiple citation. The ID value has to start with the letters "IM" as a sort of sanity check.
- ❷ This and the following `xref` elements define the actual references that comprise the multiple citation.

Note: The sequence of the `xref` elements that encode the actual references *may* be important. Depending on the bibliography style used for the document transformation, the references may be displayed in the sequence as they were entered, or they may be rearranged according to the sequence of the bibliographic entries in the finished bibliography.

Keep also in mind that all attribute values must be in uppercase for the same reasons as stated above.

The corresponding TEI citation is a little bit simpler:

```
<seg type="REFDBCITATION" part="N" TEIform="seg">
<ptr type="MULTIXREF" targOrder="U" target="IMTHEFIRST" TEIform="ptr"/>❶
<ptr targOrder="U" target="ID1-X" TEIform="ptr"/>❷
<ptr targOrder="U" target="LITIBP-ID21-X" TEIform="ptr"/>
<ptr targOrder="U" target="ID5-X" TEIform="ptr"/>
</seg>
```

- ❶ This is the additional `ptr` element which is mandatory in multiple citations. The element must have a `type` attribute with the value `MULTIXREF`. You also have to provide an unique value for the `target` attribute. This specifies the ID value that will be used in the corresponding element in the `refdb`-generated bibliography. The ID string has to start with "IM". In contrast to DocBook elements, there is no way to specify where a link should point to. The `refdb` XSL stylesheets will use the first bibliographic entry referenced in a multiple citation as the link target.
- ❷ This and the following `xref` elements define the actual references that comprise the multiple citation.

10.3.1.3. Generate the bibliography

Unless you have good reasons not to do so, you should use the `runbib` shell script to generate the bibliography. This script greatly simplifies this task and offers a common interface for all supported document types. The following subsection will explain the use of this script. If you like to do it the hard way (or if you want to peek under the hood) you'll find a few explanations further down how to do this.

10.3.1.3.1. Use `runbib`

Lets assume you have a DocBook SGML document `mypaper.sgml` and want to submit it to the "Journal of Irreproducible Results". We further assume that the bibliography style for this famous periodical is stored in your database under the name "J.Irrep.Res." (see Manage bibliography styles to learn how it gets there). All your bibliography entries (at least those referenced without an explicit database name) are stored in the database `mybib`. Start the script from the directory that contains your document with the following command:

```
~$ runbib -d mybib -S "J.Irrep.Res." -t db31 foo.sgml
```

For a similar TEI XML document `bar.xml` you would run:

```
~$ runbib -d mybib -S "J.Irrep.Res." -t teix bar.xml
```

In both cases you will end up with a bibliography file (`foo.bib.sgml` and `bar.bib.xml`, respectively) as well as with a stylesheet (`J.Irrep.Res.dsl`) or a set of stylesheets (`J.Irrep.Res.fo.xsl` and `J.Irrep.Res.html.xsl`), respectively.

Note: Don't worry if you are greeted by a list of (Open)Jade errors complaining about missing elements when you first run this script on a particular document. Your document contains a number of crosslinks that point to elements that do not exist yet - you use `runbib` precisely to create these elements (you thus face a classic bootstrapping problem). As soon as the bibliography is created, these error messages should go away. Later you will only get an error message for each bibliographic entry that was added since the last time you ran `runbib`.

10.3.1.3.2. Do it the hard way

The following steps do exactly what the `runbib` script does, just with more to type. The only benefit of the hard way is that you have a chance to fiddle with the intermediate XML file which contains the list of bibliographic entries that should go into the bibliography. You can add further entries to extend the bibliography if you want to include uncited publications. The following procedure was written with a DocBook SGML document in mind, but transferring the commands to XML documents is straightforward. However, when working with XML documents there are additional steps required as outlined below.

1. Extract the list of bibliographic references

Use Jade or OpenJade with the `citations.dsl` stylesheet to create a list of the reference IDs (provide full paths as needed):

```
#~ openjade -t sgml -d citations.dsl /usr/lib/sgml/declaration/docbook-3.1.dcl foo.sgml > f
```

Be prepared for a long list of "missing ID" error messages. This is due to the fact that the elements with the IDs that the `xref` elements in the citations point to do not yet exist, they will be generated in the `refdb` bibliography output. If you process documents with more than 200 citations, you'll have to increase the maximum error limit of Jade in order to obtain all IDs the first time. After the first complete pass (including the steps outlined below), Jade will only complain about any additional citations that you have inserted since the last run.

The output is a simple XML file that contains the information about all `citation` and `xref` elements with their relevant attributes. It is absolutely legal to extend this file with additional citation elements to specify references which are not cited but nonetheless should appear in the bibliography.

Unfortunately, both Jade and OpenJade don't get that Doctype line quite correct. Both forget to insert a space between the public and the system identifier, thus leaving you with a not well-formed document. Fire up your favourite editor and fix this line manually (insert a space between the two consecutive quotation marks on line 2).

If you edit this intermediate XML file (that is, if you do more than just fixing the Doctype line), you should make sure that the result is still valid according to the CitationList XML DTD. `refdb` uses a non-validating parser to read this file so deviations from the DTD may slip through undetected and may have undesired consequences. The intermediate XML file carries the `SYSTEM` identifier of the CitationList XML DTD in the document type declaration. You may have to adapt the stylesheet `citations.dsl` to use the correct path for your local system.

The following command line can be used to validate the document with `(o)nsgmls` (change the path to the XML declaration as necessary):

```
~$ onsgmls -wxml -s /usr/lib/sgml/declaration/xml.dcl foo.id.xml
```

2. Create the bibliography file

```
~$ refdbbib -d mybib -S "J.Irrep.Res." foo.id.xml > foo.bib.sgml
```

This assumes that your reference database is called "mybib" and that you try to publish your paper in a journal that accepts the style with the name "J.Irrep.Res.".

In addition to the bibliography file, `refdbbib` will also create a DSSSL script containing the style specification. This file is a customized driver file for the RefDB-DocBook driver files and provides a couple of variable values specific for the given bibliography style.

3. Post-processing

This step is only required for XML documents. First we have to bring the stylesheets into shape, and if it is a TEI document, we'll also have to transform the bibliography file itself.

`refdbbib` creates a general-purpose XSL stylesheet which we need to turn into one FO and one HTML stylesheet. Create two copies of the file. If the stylesheet was e.g. `J.Biol.Chem.xsl`, you need one copy named `J.Biol.Chem.fo.xsl` and one copy named `J.Biol.Chem.html.xsl`. Scan the files for an import statement whose `href` attribute is surrounded with two "`<!-- REFDBSTYLESHEET`

-->" comments. The value of this attribute must be set to the full path of the corresponding original stylesheet (DocBook FO or HTML, or TEI FO or HTML).

If you're working on a TEI XML document, you'll have to transform the bibliography file itself. This is a DocBook SGML document and can be transformed easily with Jade/OpenJade and the `bibdb2tei.dsl` stylesheet.

10.3.1.4. Transform the document

Finally you can transform the document to create printable or HTML output. In order to get the formatting of the citations and bibliography entries right you have to use the `refdb` driver files for the DocBook or TEI stylesheets.

In addition to the general modifications of these driver files we'll have to apply modifications specific for the particular reference style. Therefore you have to specify the DSSSL or XSL style specification file that was created in the previous step. For your convenience it is recommended to use the supplied `refdbjade` and `refdbxml` scripts for DSSSL and XSL transformations, respectively, which were designed for this task:

```
~$ refdbjade -t html -s J.Irrep.Res.dsl foo.sgml
```

```
~$ refdbxml -t pdf -s J.Irrep.Res.fo.xsl bar.xml
```

If you want to change the bibliography style of your document, all you need to do is to rerun **runbib** and **refdbjade** or **refdbxml** with the new parameters. No changes to your DocBook source are necessary.

Note: If you want to create a bibliography for each part of a book or for each chapter, the procedure is not much different. The simplest approach is to keep the parts or chapters in individual files and process these individually as described above for the whole document. You'll get several bibliography files that you can include into the corresponding document source files.

10.3.1.5. The `refdbnd` shortcut

Now that you know all necessary steps to process SGML and XML documents, it's about time to reveal that there is a simple shortcut if you can live with some minor restrictions. The `refdbnd` script helps you to start new SGML or XML projects and sets up a Makefile to process your document.

Start the script in a clean subdirectory by typing **refdbnd**. The script will start in interactive mode and ask a couple of questions. You'll have to specify the basename of your project, the SGML or XML document type declaration you'd like to use, the top-level element, the `refdb` database that holds the references which you intend to cite, the name of the bibliography style to be used with this document, the character encoding, and the name of a CSS file to be used with (x)html output. The script will then create a file `<basename>.short.[sgml|xml]`. The ".short" reminds you that the Makefile assumes you will be

using the short notation for citations. It will also create a Makefile which is set up to perform the necessary steps to create all sorts of available formatted output.

Once you have written your document, including a few citations and a reference to the external bibliography file as explained in the previous sections, you can use the Makefile to process your document. You may know how to use Makefiles anyway, but if not, here are the main properties:

- A Makefile is an input file for the program **make**. If your present working directory contains a file called `Makefile` and you just run **make**, the program will process that file.
- Makefiles define one or more targets. A target defines which kind of output you want to create. In order to run a specific target you pass this target as an argument to **make**.
- Makefiles define dependencies. Unlike simple shell scripts that usually run a complex series of commands from start to end, **make** checks for each target individually whether one of the files or targets it depends on is outdated. This way, only the minimum number of required processing steps will run in order to create the desired output.

The Makefiles created by `refdbnd` offer the following targets:

pdf

This target generates a PDF file from your source document. PDF is a widely accepted document format with free viewers for essentially all current operating systems. Be aware that not all FO processors (used in transforming XML documents) offer PDF output.

html

This runs all required commands to create HTML output, viewable with any web browser. Depending on your local setup, the output will be chunked into a collection of HTML files.

rtf

This target generates a Rich Text Format (RTF) file. This plain text format is sort of a word processor interchange format understood by most current word processors, including MS Word, WordPerfect, and OpenOffice/StarOffice. Be aware that not all FO processors offer RTF output.

ps

This target is only available for SGML documents. It will create a Postscript document from your source. Postscript is the universal document format on Unix systems and can be printed directly on Postscript printers. Viewers are available for all current operating systems.

The Makefile also offers a few more targets. For each of the above targets there is a corresponding '`<target>dist`' target which creates a `.tar.gz` archive of the output document, along with its associated CSS stylesheet if applicable. The target '`all`', which is also the default if you don't specify a target to **make**, builds all available output formats. Accordingly, the target '`dist`' creates all archives. And finally, the target '`clean`' removes all intermediate files and returns your directory to the original state.

For example, to create a formatted PDF document `<basename>.pdf` from your `<basename>.short.sgml` file you'd type **make pdf**. **make** will first convert the short-style citations

to the full style using the **refdbxp** tool. Then it will generate the bibliography and the stylesheet driver files by running **refdbbib**. Finally it will run the **refdbjade** script to create the PDF output.

The **refdbnd**-generated Makefiles should be sufficient for the average document. However, feel free to modify them in order to adapt them to specific needs. For example you can specify a different style in order to switch your output to a different citation and bibliography style. **make** also allows you to override variable settings on the command line. E.g. if you want to output your document using a different bibliography style without making it the permanent default, invoke **make** like this:

```
~$ make pdf stylename="Eur.J.Pharmacol."
```

Note: You'll have to remove all intermediate files by running **make clean** first before you can switch to a different bibliography style.

10.3.2. Other SGML or XML document types

While **refdb** works out of the box with DocBook SGML/XML and TEI XML documents, it is by no means limited to these document types. The only native bibliography export format is DocBook (in a form suitable for both SGML and XML documents). TEI bibliographies are actually generated from this output with a SGML-to-XML transformation. The DocBook output has a sufficient granularity to allow this and possibly other transformations. If you want **refdb** to work with other document types, you have to do the following:

- Either extend `citations.dsl` or create an additional stylesheet suitable to extract a list of citations and references conforming to the CitationList XML DTD.
- Provide a DSSSL stylesheet to transform the DocBook SGML bibliography output to your target document type.
- Modify your DSSSL or XSL stylesheets (or better, provide suitable driver files) to make use of the extended formatting information both in the bibliography and in the **refdb**-created driver files.

10.3.3. LaTeX/BibTeX

refdb integrates quite nicely with the LaTeX/BibTeX system. If you previously used a flat text file to store your BibTeX references, you will notice that there is only one additional command to run when you process your source document. Instead of keeping all of your references in a text file, **refdbbib** will retrieve only the required references from the SQL database and store them in an intermediate text file.

1. Prepare the document

Use the LaTeX commands **cite** and **nocite** to include the references as usual. The extended commands from the **natbib** package should work as well. All these commands take an identifier for

the reference as an argument. These reference definitions can come in two flavours just like in DocBook documents: Either you use the same database for all references in the text. Then you just specify the ID of the reference and tell the processing application which database to use. Or you specify the database name with each reference. In this case, you can pull the references from different databases in the same document. The two versions look like this:

```
\cite{IDMiller1999}
\cite{litibp-IDMyers2001}
```

The first version cites the reference with the identifier “1” in the database passed to the processing application as an argument. The second form cites the reference with the identifier “2” in the database “litibp”.

The LaTeX **\bibliography** command takes as an argument the name of the intermediate bibliography file without the extension. A simple choice would be the basename of your LaTeX document.

Note: Keep in mind that even if you pull references from different refdb databases, you still need to specify only one reference database in your LaTeX document as refdb consolidates all cited references into one bibliography file.

2. Create the auxiliary file

Run the latex interpreter with the basename of your document (`foo.tex`) as an argument:

```
#~ latex foo
```

latex will create, among other files, `foo.aux`. latex stores all sorts of information in these auxiliary files for later use in subsequent runs. The interesting part for us is the list of citations.

3. Create the intermediate bibliography file

Now refdb enters the stage. We process the auxiliary file to create a BibTeX bibliography tailored to our document. Either we do it manually:

```
#~ refdbbib -d mybib -S name -t bibtex foo.aux > foo.bib
```

Or we use the **runbib** shell script:

```
#~ runbib -d mybib -S name -t bibtex foo
```

Remember that the basename of the file that receives the bibliographic information (`foo.bib` in our example) must match the name given in the **\bibliography** command in the LaTeX document.

The resulting bibliography file will contain all references that were requested from the LaTeX document. If you add more citations to this document, you have to run refdbbib again to update the intermediate bibliography file (it won’t hurt if you *remove* citations from your LaTeX document, though).

Note: For the sake of consistency with **bibtex**, it is possible (though not necessary) to specify the auxiliary file without the `.aux` extension (`foo` in the above example).

4. Run bibtex

From here, everything runs as you are used to from LaTeX/BibTeX:

```
#~ bibtex foo
```

5. Run latex

Run latex on your LaTeX document at least twice to get all references right:

```
#~ latex foo && latex foo
```


IV. Reference manual

Chapter 11. reftdb: the application server

11.1. Overview

reftdb is the reftdb application server, i.e. a program that runs in the background and handles the requests from the reftdb clients. This is the program that directly interacts with the database server and does most of the serious work.

reftdb can be configured at startup with the configuration file `reftdbdrc` or by passing command-line options. The latter override the corresponding settings in the configuration files. While reftdb is running, most of the parameters can be changed remotely with reftdba (see the reftdba command **confserv**), but to make changes permanent you'll have to edit the configuration file or the script line that starts reftdb.

Warning

In the current implementation, the remote administration via reftdba uses a fairly indirect access control. If this is a concern, you should not enable remote administration (it is disabled by default). You can use the command **reftdbctl reload** instead to cause reftdbd to reread its configuration file while running. Then the usual access controls for editing the configuration file and for sending a signal to a process apply.

You can run reftdbd as a standalone application or as a daemon. The main difference between the standalone process and a daemon process is the fact that the daemon is no longer associated with a particular terminal. This means that no output to stdout or stderr will ever show up on a terminal and that the process will keep running even if the terminal that started the process terminates. This is good if you want it running unattended, similar to a web server or a ftp server. On the other hand it can be handy for debugging purposes to directly get log output (or additional `printf()` output) on a terminal and to use the standard ways of process control, e.g. to kill reftdbd with **Ctrl-C**. The `-s` command line switch lets reftdbd run as a normal application for exactly this purpose.

The above holds true for almost any Unix-like operating system. On Windowsish operating systems the story is somewhat different. First of all, the Win95/98/ME family does not support daemons at all. WinNT/2000 do support daemons (they're called "services" here just for the sake of having a different name for the same thing). The simplest way to run reftdbd as a NT service is to use a service installer as described in the Installation chapter. In this case you should run reftdbd as a standalone application (i.e. include the `-s` switch), as the installer will take care of all the magic. Another possibility is to start reftdbd from Cygwin inetd. In that case you'd run reftdbd as a daemon again, but this is currently beyond the scope of this manual.

You can run reftdbd as a daemon either from the command line, i.e. start it manually when you need it, or you can start the daemon at the system start. For the latter, refer to the Installation chapter.

11.2. Starting refdbd

This section explains how to control refdbd from the command line. The easiest and strongly recommended way is to use the provided control script that will be explained in the first subsection. If you need further control, you can use the command line of refdbd directly as explained in the second subsection.

11.2.1. Using the control script

The refdb package includes the `refdbctl` script which is installed in `/usr/local/bin`. This is a simple shell script to start, stop, restart, and reconfigure refdbd from the command line. The very same script is also used if you have instructed `init(8)` to start and stop refdbd.

`refdbctl` accepts the following arguments:

start

refdbctl start will try to start the daemon. refdbd will be called without any command-line parameters so all configuration parameters should be set in `refdbdrc`. The script prints a diagnostic message whether or not the daemon was successfully started. If the daemon is already running when you issue this command, the script will *not* try to start another process.

stop

refdbctl stop will try to stop the daemon. This is done by sending the process the `TERM` signal. The script prints a diagnostic message whether or not the daemon was successfully stopped. If the daemon is not running when you issue this command, the script will print a diagnostic message.

reload

refdbctl reload will cause the daemon to reload its configuration file by sending the process the `HUP` signal. The script prints a diagnostic message whether or not the daemon configuration was successfully reloaded. If the daemon is not already running when you issue this command, the script will *not* try to start it.

restart

refdbctl restart will terminate the daemon and start it again. If the daemon is not yet running, this command will not attempt to start it.

11.2.2. Start refdbd directly

Calling refdbd directly allows you to pass configuration parameters on the command line. Start refdbd with the command:

```
refdbd [-b dbs-port] [-d default-database] [-D dbserver] [-e log-destination] [-E
encoding] [-h] [-i IP-address] [-I] [-k] [-K] [-l log-level] [-L log-file] [-p port] [-P
```

```
PID-file] [-q] [-r] [-s] [-S note-share-mode] [-T time] [-U] [-v] [-V] [-y confdir] [-Y
libdbidir]
```

The command-line switches are:

-b *dbb-port*

Set the port on which the database server listens for incoming connections. The default is 3306 for MySQL and 5432 for PostgreSQL. This option does not apply if you use SQLite as your database engine.

-d *default-database*

Set a default database to be used for all client queries that do not specify a database.

-D *dbserver*

Select the database server. Currently the values *mysql*, *pgsql*, and *sqlite* are supported to select MySQL, PostgreSQL, and SQLite, respectively.

-e *destination*

This specifies the destination of the log information. If destination is 0 or "stderr", the log output is sent to stderr. This should only be used for debugging purposes when *refdbd* is not run as a daemon.

If destination is 1 or "syslog", the syslog facility of the system is used. syslog has to be configured properly to accept *refdb*'s log output. Consult the *syslogd(8)* man page how to achieve this.

If destination is 2 or "file", a custom log file as defined by the *-L* switch is used instead. If this log file cannot be written to, *refdbd* falls back to using syslog.

-E *encoding*

Select the default character encoding for new reference databases. This feature is currently supported only by PostgreSQL (and only if PostgreSQL was built with the *--enable-multibyte* option). See the PostgreSQL manual for a full list of supported encodings. As of this writing, SQL_ASCII, EUC_JP, EUC_CN, EUC_KR, EUC_TW, UNICODE, MULE_INTERNAL, LATIN1 through LATIN5, KOI8, WIN and ALT are available. You can override this default by using the *-E* option of the **createdb** command.

-h

Displays help and usage screen, then exits.

-i *IP-address*

For external database servers, set the IP address of the box which is running the database server. Instead of the IP address you can also specify the hostname as long as it can be properly resolved by your system. If the database server runs on the same box as *refdbd*, use the string "localhost".

Note: If "localhost" does not seem to work, try specifying the real IP address of the box instead - some database client libraries refuse to use TCP/IP for local connections which may cause mayhem on particular systems.

For embedded database engines, this option sets the directory which contains the database files. The default is `/usr/local/share/refdb/db`.

Note: SQLite on Cygwin has a bug which prevents the use of absolute paths. Use a relative path instead by leaving out the leading slash. This assumes that `refdbd` is started from the root directory. The start script `refdbctl` does this automatically.

-I

Set this switch to allow remote connections to `refdbd`. Otherwise only connections from localhost (127.0.0.1) will be answered.

-k

`refdb` stores up to four names (full name, official abbreviation, and two user-defined abbreviations) of each periodical. These synonyms are shared by all references that use one of these periodical names. As it sometimes requires some effort to get at the synonyms (public reference data often contains only the official abbreviation), it is desirable to keep these synonyms even if you remove the last reference that uses a particular periodical name. If you use the `-k` option, the synonyms will not be removed from the database and will be available immediately if you add a new reference using that particular periodical name.

-K

If this option is used, `refdbd` will run an automatic keyword scan each time you add or update references. `refdbd` will scan the title fields and the abstract field of the modified references for any keywords already present in the database but not in the particular reference. This increases the usability of keywords in queries. There is no speed decrease for the user interaction as the keyword scan is performed in the background. See also the related `refdba` command `scankw` which performs a more thorough manual keyword scan.

-l level

Set the log level to a value between 0 and 7 or to a string value as described in log level definitions. 0 means that only critical log messages will be logged, while a value of 7 means that every log message will be logged. Set level to -1 to disable logging.

Caution

Use log level 7 with caution and only for debugging purposes. Every SQL command and a plethora of other messages will be logged. This slows down the application server considerably and has the potential to grind your server to a halt by completely clogging the hard drive :-).

-L file

This switch specifies a custom log file (full path please). This will only be used if the -e switch is set accordingly.

Warning

If you start refdbd as a regular user, you may not have write access to the default log file. In this case, use this option to specify a file that you can write to, e.g. /home/username/refdbd.log.

-p port

Set the port on which refdbd listens for incoming connections. The default is 9734.

-P

Specify the full path of the file that refdbd writes its process ID to. This PID simplifies stopping and reconfiguring the application server from the command line. The default value is /var/log/refdbd.pid.

Warning

If you start refdbd as a user, you may not have write access to the default PID file. In this case, use this option to specify a file that you can write to, e.g. /home/username/refdbd.pid.

-q

Start without reading the configuration file. Useful for debugging purposes.

-r

Enables remote administration via refdba.

-s

Starts as a standalone application, not as daemon.

-S

Set the default extended note share mode to either *public* or *private*. This setting affects the accessibility of extended notes if they do not explicitly carry a *share* attribute. See the section about notes sharing for more information.

-T *time*

Set the timeout for client/application server dialogue in seconds.

-U

This switch causes *refdbd* to automatically uppercase all citation keys of newly added references. This makes it more convenient to work with SGML bibliographies.

-v

Prints version and copyright information, then exits.

-V

Switches to verbose mode. To be honest, currently this doesn't make much of a difference.

-y

Specify the directory where the global configuration files are

Note: By default, all *refdb* applications look for their configuration files in a directory that is specified during the configure step when building the package. That is, you don't need the `-y` option unless you use precompiled binaries in unusual locations, e.g. by relocating a rpm package.

-Y

Specify the directory where the *libdbi* drivers are

Note: By default, *libdbi* (the database abstraction library used by *refdb*) looks for its driver files in a directory that is specified during the configure step when building the package. That is, you don't need the `-Y` option unless you use precompiled *libdbi* binaries in unusual locations, e.g. by relocating a rpm package.

11.3. The refdbdrc variables

Table 11-1. refdbdrc

Variable	Default	Comment
refdblib	/usr/local/share/refdb	The path of the directory containing shareable refdb files like DTDs, HTML templates etc. Actually, most of the files are in subdirectories of refdblib.
dbsport	3306	The port on which the database server listens. Use either 3306 or 5432 for MySQL and PostgreSQL, respectively. This variable is ignored if you use SQLite as your database engine.
dbserver	mysql	The database server you want to connect to. Use one of mysql, pgsql, or sqlite to select MySQL, PostgreSQL, or SQLite as your database engine, respectively.
dbpath	/usr/local/share/refdb/db	The directory that contains the database files of an embedded database engine. Leave out the leading slash if you use SQLite on Cygwin (see note above).
logdest	2	The destination of the log information. 0 = print to stderr (for debugging only, don't use when running as a daemon); 1 = use the syslog facility; 2 = use a custom logfile. The latter needs a proper setting of logfile.
logfile	/var/log/refdbd.log	The full path of a custom log file. This is used only if logdest is set appropriately. If you start refdbd from the command line as a regular user, you should specify a file that you have write access to (you may not be allowed to create /var/log/refdbd.log or write to this file as a regular user).

Variable	Default	Comment
loglevel	6	The log level up to which messages will be sent. A low setting (0) will notify you only in case of a meltdown, whereas a high setting (7) allows all messages including debug messages (this is <i>a lot</i>). -1 means nothing will be logged.
pidfile	/var/log/reftdbd.pid	The file reftdbd writes its process ID to. If you start reftdbd from the command line as a regular user, you should specify a file that you have write access to (you may not be allowed to create /var/log/reftdbd.pid or write to this file as a regular user).
port	9734	The port on which reftdbd listens. The server and all clients that are supposed to connect to it must agree on the same port. Obviously, this option allows to run several instances of reftdbd on the same box if there is a good reason to do so. In this case you should also use separate log and pid files.
remoteadmin	f	Set this to 't' to allow remote administration via reftdba. Be aware that this is a security risk.
serverip	localhost	The IP address or hostname of the machine where the database server runs. Use the default (localhost) address if the database server and reftdbd run on the same machine.
timeout	180	The timeout in seconds. After this time has elapsed, a stalled connection is taken down. Increase this value if you encounter frequent timeout errors due to high network traffic.

Variable	Default	Comment
keep_pnames	t	Set this to 't' to keep periodical names and synonyms if you remove references. If set to 'f', the names will be removed from the database if the last reference using these names is deleted.
db_encoding	(none)	Specify the default character encoding for new refdb reference databases. If the database server supports this feature (currently only PostgreSQL does), all new databases will use this encoding unless a different one is specified with the createdb command.
in_encoding	ISO-8859-1	Specify the default character encoding for RIS data being added to databases.
dbi_driverdir	(none)	Specify the directory containing the libdbi driver files. As mentioned above, this is only necessary if you use precompiled libdbi binaries in funny locations.
keyword_scan	t	Set this to 't' to allow an automatic keyword scan after references are added or updated. 'f' will switch off this feature.
upper_citekey	f	Set this to 't' to uppercase all citation keys of newly added references (this makes sure they work with SGML bibliographies).
share_default	public	Whether ("public") or not ("private") to share extended notes between users by default. See the section about notes sharing for more information.
remoteconnect	f	Set this to 't' to allow remote connections to refdbd. By default, refdbd accepts only local connections due to security concerns.

Chapter 12. refdba: the administrative client

12.1. Overview

The following tasks can be performed with refdba:

- Create and delete databases.
- Add, remove, and retrieve bibliography styles.
- Add and remove users
- Add and remove reserved journal words
- View statistics and connection information.
- Configure the application server while running

12.2. Starting refdba

To start an interactive refdba session, use the command:

```
refdba [-c pager-command] [-e log-destination] [-h] [-i IP-address] [-l log-level] [-L log-file] [-p port] [-q] [-T time] [-u name] [-v] [-V] [-w password] [-y confdir]
```

The meaning of the command-line switches is explained in the section common command-line options.

Unless you specify a password in a configuration file or on the command line, refdba will ask you for a password before you can proceed. When refdba runs, you are greeted with the prompt:

```
refdba:
```

To start refdba in batch mode, use the following command:

```
refdba [-c pager-command] {-C command} [-e log-destination] [-i IP-address] [-l log-level] [-L log-file] [-p port] [-q] [-T time] [-u name] [-w password]
```

refdba will execute the internal command whose name is specified with the -C option and exit. You can specify all command-line options that the internal command may need on the refdba command line.

The exit code is 0 if all went fine. It will be 1 if the command (when run in batch mode) or the last command (when run in interactive mode) returned an error, or if there was a general error condition during startup like a lack of available memory.

12.3. The reftdbarc variables

Table 12-1. reftdbarc

Variable	Default	Comment
logfile	/var/log/reftdb.log	The full path of a custom log file. This is used only if logdest is set appropriately. If you start reftdb from the command line as a regular user, you should specify a file that you have write access to (you may not be allowed to create /var/log/reftdb.log or write to this file as a regular user).
logdest	2	The destination of the log information. 0 = print to stderr (this is mainly intended for debugging, as it may visually interfere with command output); 1 = use the syslog facility; 2 = use a custom logfile. The latter needs a proper setting of logfile.
loglevel	6	The log level up to which messages will be logged. A low setting (0) allows only the most important messages, a high setting (7) allows all messages including debug messages. -1 means nothing will be logged.
pager	stdout	The command line of a pager that accepts the output of reftdb on stdin to allow scrolling and other nifty things. "stdout" sends the data to stdout.

Variable	Default	Comment
passwd	*	The password which is used for authentication with the database server. It is potentially evil to store unencrypted passwords in disk files. At least make sure that the configuration file is not readable for anyone else. The default setting causes refdba to ask for your password interactively.
port	9734	The port on which refdbd listens. Change this for all clients and the server if this value interferes with another program using this port.
serverip	127.0.0.1	The IP address or hostname of the machine where refdbd runs. Use the default (localhost) address if the clients and refdbd run on the same machine.
timeout	180	The timeout in seconds. After this time has elapsed, a stalled connection is taken down. Increase this value if you encounter frequent timeout errors due to high network traffic or refdbd overload.
username	login name	The username which is used for authentication with the database server. This may be different from the login name of the user.
verbose	f	Set this to t if you prefer verbose error messages.

12.4. Commands

All commands consist of a single word which specifies the command. This may be followed by arguments and/or switches. The general syntax rules of the getopts library apply.

12.4.1. addstyle

12.4.1.1. Synopsis

```
addstyle [-c] [-h] [-o | -O] {style-file...}
```

12.4.1.2. Description

Adds one or more bibliography style specifications from the input file(s).

Use the `-c` option to specify a command that will receive the output instead of the default pager. This may be a different pager, any command that takes input on stdin, or the string “stdout” to send the data to stdout without using a pager.

The `-h` option displays the online help about the **addstyle** command.

Use the `-o` and `-O` options to write or append the output to a file, respectively. This may be useful if you want to add a whole bunch of style files with a single command.

All other arguments are interpreted as the names of files containing style specifications.

12.4.1.3. Example

```
refdba: addstyle j.biol.chem.xml pharmacol.rev.xml
```

This will add the style specifications contained in the files `j.biol.chem.xml` and `pharmacol.rev.xml` to the bibliography style database.

12.4.2. adduser

12.4.2.1. Synopsis

```
adduser {-d database} [-h] {-H host-IP} [-N newpassword] {-f file | username}
```

12.4.2.2. Description

Grants access rights to a refdb database to the given users. Specify the database with the `-d` option.

Note: If a user is not yet known to the database server, refdb will create an account with the default access rights (=none). If you do not specify a password for the new user with the `-N` option (see

below), the user will have access to the database server with the default password "refdb". In most cases this is not a good thing.

A new user will automatically get access to the internal refdb database refdb.

Some database engines like SQLite do not support access control. The **adduser** command is not supported with these engines and will just return an explanatory message.

The **-h** option displays the online help about the **adduser** command.

The **-H** option takes a hostname as an argument. This is the name of the host the refdb application server runs on. If it runs on the same machine as the database server, you may specify localhost as hostname. Use **"%"** as hostname to allow access from all addresses except localhost. Otherwise, the hostname argument can be either a hostname, an IP address, or a subnet that specifies one or more computers to allow access from. You can add the same user several times with different hostnames.

Note: This option is only supported by MySQL. It is ignored if you use PostgreSQL as your database server. Please see the PostgreSQL documentation for help on how to manipulate host-based access control with the `pg_hba.conf` file.

Use the **-N** option to specify the password for a new user. The password is encrypted before transferring it to the application server. If the user already exists, his password will be changed accordingly.

All other arguments are interpreted as usernames. In addition to naming users explicitly, you can use the **-f** option to specify a file which contains a whitespace-separated list of usernames. If neither a username argument nor an input file is specified, refdba attempts to read a whitespace-separated list of names from stdin. To force refdba to read from stdin *in addition to* explicitly named users, use the **-f stdin** option.

12.4.2.3. Examples

```
refdba: adduser -d db1 -N newpassjim
```

This will grant access to the database `db1` for the new user `jim`. `refdbd` runs on the same computer as the database server (if you leave out the **-H** option, localhost is assumed). "jim" will have to provide "newpass" as a password when starting one of the refdb clients.

```
refdba: adduser -d db1 -H mono.mycomp.com jim jane
```

This will grant access to the database `db1` for the users `jim` and `jane`. `refdbd` runs on the computer with the name "mono.mycomp.com". If "jim" and "jane" are already known to the database server, they will keep their existing passwords. If not, they will have to use the default password "refdb".

12.4.2.4. Alternatives on sites with restricted database server access

If you as the *refdb* administrator do not have GRANT permission on your database server, the **adduser** command is bound to fail. As a security-minded person your database administrator might refuse to run *refdba* regardless of how often you ensure him it doesn't contain malicious code. He'll want to do it the hard way, and this is what he needs to do:

- If you use MySQL as your database server, each new user needs at least entries in the *mysql.user* and *mysql.db* tables. Your database administrator might have set up his own rules, but in general the *mysql.user* table should grant no privileges to the user, whereas the *mysql.db* table should grant INSERT, SELECT, UPDATE, DELETE permissions to each user for the *refdb* database and SELECT, INSERT, UPDATE, DELETE, CREATE, DROP privileges for each reference database the user should have access to. Make sure to mention that the Host field in *mysql.user* must contain the name or address of the box that runs *refdbd*, which is not necessarily identical with the workstation of the user.
- If you prefer PostgreSQL instead, things are a little simpler. When you create a *refdb* database, a new group will be created to manage access to this database. All your database administrator needs to do is to add the new user to the groups *refdbuser* (granting access to the common *refdb* database) and *<dbname>user*, where *<dbname>* is the name of the reference database the user should be allowed to access.

12.4.3. addword

12.4.3.1. Synopsis

```
addword [-h] {-f file | word}...
```

12.4.3.2. Description

Most bibliography styles use standardized abbreviations of the journal names. Most data sources specify these abbreviations without dots, as in "Mol Cell Biol". If the words are to be abbreviated with dots (as in "Mol. Cell Biol.") in the bibliography, *refdb* needs to know which tokens in the abbreviated name are indeed abbreviated (e.g. "Mol."), and which are full words (e.g. "Cell"). To this end, *refdb* keeps a list of reserved words which are known not to be abbreviations of something else. *refdb* ships with a fairly complete list of such words, but if you detect errors or omissions, the **addword** command comes in handy.

The *-h* option displays the online help about the **addword** command.

All other arguments are interpreted as reserved words. In addition to listing the words explicitly, you can use the *-f* option to specify a file which contains a whitespace-separated list of reserved words. If neither

a word list nor an input file is specified, `refdba` attempts to read a whitespace-separated list of words from `stdin`. To force `refdba` to read from `stdin` *in addition to* explicitly listed words, use the `-f stdin` option.

Note: `refdb` will convert all reserved words to uppercase internally, so it does not matter in which case you provide these words.

12.4.3.3. Example

```
refdba: addword -f wordlist FOO BAR
```

This will add all reserved words in the file `wordlist` as well as the words "FOO" and "BAR" to the list of reserved words.

12.4.4. `confserv`

12.4.4.1. Synopsis

```
confserv {command} [value]
```

12.4.4.2. Description

Configures the application server while it is running and does some tricks with the `refdb` helper databases as well. Some of the commands modify variables that can be set as command line arguments or with the `init` file. See *Running the `refdbd` daemon* for more information about these variables.

Note: This command will only reconfigure `refdbd` transiently. All changes are lost when the application server is restarted. To make permanent changes to the configuration, edit the `init`-file or change the command-line parameters in the script that starts `refdbd`. Please note also that remote administration must be enabled for this command to work.

The following commands are available:

stop

Stops the application server.

Note: This command affects only the `refdbd` parent process. Any children that may be currently serving clients will continue to do so until they are done.

ping

Checks whether the application server is still alive and well. If this is the case, it will report the process IDs of the child that handles your query and of the parent. If not, the connection will time out with no response.

serverip *value*

Sets the database server IP address to *value*.

timeout *value*

Sets the timeout in seconds to *value*.

logdest *value*

Sets the destination of log output to *value*. Possible values are 0 (stderr), 1 (the system syslog facility), 2 (a private log file as defined by *logfile*).

logfile *value*

Sets the filename of the log file to *value*.

loglevel *value*

Sets the maximum level of messages to be logged to *value*. 0 means that only critical errors will be logged, 7 means that all messages including the extremely verbose debug messages will be logged. -1 disables logging completely.

12.4.4.3. Example

```
refdba: confserv loglevel 7
```

This will set the log level to 7. This temporary change will only be effective until *refdbd* is restarted.

12.4.5. createdb**12.4.5.1. Synopsis**

```
createdb [-E encoding] [-h] {dbname...}
```

12.4.5.2. Description

Creates a new database with the name *dbname*. Several databases may be specified in a single call of this command.

The `-E` option selects a character encoding for the new database. This is currently only supported by MySQL and PostgreSQL. If you use a different engine, this option is ignored. Please see the documentation of your database engine installation for available encodings. The value passed with the `-E` option should be the IANA (<http://www.iana.org>) encoding name. If you do not use this option, the new database will use the default encoding of the database server unless your `refdbdrc` configuration file sets a default with a `"db_encoding"` entry.

Tip: Prepend a constant string like `"rd"` to all `refdb` database names. This speeds up retrieving `refdb` databases with the `listdb` command. Use a simple regular expression like `"rd%"`.

12.4.5.3. Example

```
refdba: createdb db1 -E UTF-8 db2
```

This will create the databases `db1` and `db2` with the character encoding UTF-8.

12.4.5.4. Using SQL scripts to create databases

`refdb` contains two plain-text SQL scripts (installed in `/usr/local/share/refdb/sql`) to create database tables just like the `createdb` command does. These scripts are preferable to the command in these cases:

- You do not have database administrator permissions and have to ask your admin to create the databases for you. Your admin might prefer to run the script as he can easily find out what it is going to do.
- You want to integrate `refdb` with an existing or a custom database system. In that case you want the `refdb`-specific tables in an existing database in addition to non-`refdb` tables.

The following procedures are equivalent to running the `createdb` command. If you want to add the tables to an existing database, please adapt the scripts and/or the procedures accordingly.

- If you're running MySQL, use the following commands (provide additional options like username and password as required):

```
#~ mysql -e "CREATE DATABASE dbname"
```

```
#~ mysql dbname < empty.mysql.dump
```

- If you're using PostgreSQL, the following sequence should work (again, provide additional options like username and password as required):

```
#~ sed 's/refdbtest/dbname/g' < empty.pgsql.dump.in > empty.pgsql.dump
```

```
#~ psql template1 < empty.pgsql.dump
```

The `empty.pgsql.dump.in` script contains the commands to create a database and to set appropriate access rights for a new group of database users. Therefore it is a good idea to replace the string "refdbtest" with the intended name of your new database. The `sed` command in the first line does just this. You may also edit a few more things, like the encoding. The second command actually creates the database, a new group, grants privileges to this group, and creates all necessary tables and sequences. `template1` is a PostgreSQL system database. The `psql` command requires the name of an existing database as an argument, but in this case you could use any other existing database just as well.

12.4.6. deletedb

12.4.6.1. Synopsis

```
deletedb [-h] {dbname...}
```

12.4.6.2. Description

Deletes the database with the name *dbname*. Several databases may be specified in a single call of this command.

The `-h` option displays a brief usage message and returns to the prompt.

Caution

The database structure and the data will be gone, really gone, so be careful with this command. Think twice and, if in doubt, at least make a backup first to avoid extensive hairpulling.

12.4.6.3. Example

```
refdba: deletedb db1 db2
```

This will delete the databases `db1` and `db2`.

12.4.7. deletestyle

12.4.7.1. Synopsis

deletestyle [-h] {*unix-regexp*}

12.4.7.2. Description

Deletes the bibliography styles whose names match the Unix regular expression *unix-regexp*.

Note: Some database engines, like SQLite, do not support Unix-style regular expressions. Use SQL regular expressions instead.

The -h option displays a brief usage message and returns to the prompt.

12.4.7.3. Example

```
refdba: deletestyle J\.*
```

This will delete all bibliography styles that start with “J.”.

12.4.8. deleteuser

12.4.8.1. Synopsis

deleteuser {-d *database*} [-h] {-H *host-IP*} {-f *file* | *username*}

12.4.8.2. Description

Revokes access rights to a refdb database from the given users. Specify the database with the -d option.

Note: Some database engines like SQLite do not support access control. The **adduser** command is not supported with these engines and will just return an explanatory message.

refdb will only revoke the access rights to the specified database. It will revoke neither access rights to the internal database refdb, nor will it revoke database server access. You can revoke access to the internal database by specifying "refdb" with the -d option. To revoke access to the database server, please use the command line utilities of your database server.

The `-h` option displays the online help about the **adduser** command.

The `-H` option takes a hostname as an argument. Access rights will be revoked only for this particular hostname. This must be the same name that you used for a previous call to **adduser**.

Note: This option is only supported by MySQL. It is ignored if you use other database engines.

All other arguments are interpreted as usernames. In addition to naming users explicitly, you can use the `-f` option to specify a file which contains a whitespace-separated list of usernames. If neither a username argument nor an input file is specified, **refdba** attempts to read a whitespace-separated list of names from `stdin`. To force **refdba** to read from `stdin` *in addition to* explicitly named users, use the `-f stdin` option.

12.4.8.3. Examples

```
refdba: deleteuser -d -H % db1jim
```

This will revoke the access to the database `db1` for the user `jim` for all but local connections.

12.4.9. deleteword

12.4.9.1. Synopsis

```
deleteword [-h] {-f file | word}...
```

12.4.9.2. Description

This command performs the reverse operation of **addword**. The specified reserved words will be removed from the list.

The `-h` option displays the online help about the **addword** command.

All other arguments are interpreted as reserved words. In addition to listing the words explicitly, you can use the `-f` option to specify a file which contains a whitespace-separated list of reserved words. If neither a word list nor an input file is specified, **refdba** attempts to read a whitespace-separated list of words from `stdin`. To force **refdba** to read from `stdin` *in addition to* explicitly listed words, use the `-f stdin` option.

Note: **refdb** will convert all reserved words to uppercase internally, so it does not matter in which case you provide these words.

12.4.9.3. Example

```
refdba: deleteword -f wordlist FOO BAR
```

This will delete all reserved words in the file `wordlist` as well as the words "FOO" and "BAR" from the list of reserved words.

12.4.10. `getstyle`

12.4.10.1. Synopsis

```
getstyle [-c] [-h] [-o | -O] {style...}
```

12.4.10.2. Description

Retrieves one or more bibliography style specifications from the database and formats them as an XML file.

Use the `-c` option to specify a command that will receive the output instead of the default pager. This may be a different pager, any command that takes input on stdin, or the string "stdout" to send the data to `stdout` without using a pager.

The `-h` option displays the online help about the **getstyle** command.

Use the `-o` and `-O` options to write or append the output to a file, respectively.

Warning

Be careful with the append (`-O`) option. `refdb` will output the processing instructions, the doctype line, and one `CITESTYLE` element for each individually requested style. If you concatenate the results of several **getstyle** calls, the resulting XML file will not be well-formed without further processing. In order to write several styles into a single XML file, use a single **getstyle** call and list all required styles as arguments. This will output the styles wrapped in a `STYLESET` element, resulting in a valid XML file.

All other arguments are interpreted as the names of bibliography styles.

12.4.10.3. Example

```
refdba: getstyle -o j.biol.chem.xml J.Biol.Chem.
```

This will write the style specification stored under the style name "J.Biol.Chem." to the file `j.biol.chem.xml`.

12.4.11. help

12.4.11.1. Synopsis

help

?

12.4.11.2. Description

Displays a brief summary of the available commands.

12.4.11.3. Example

```
refdba: help
```

12.4.12. listdb

12.4.12.1. Synopsis

listdb [*SQL-regexp*]

12.4.12.2. Description

Lists all available databases if no argument is specified. If *SQL-regexp* is specified, only the databases matching this expression will be listed. *SQL-regexp* has to be a valid SQL regular expression.

Note: In order to tell *refdb* reference databases apart from other databases maintained by your database server, *refdbd* has to peek into each database returned by the database server. Depending on the number of available databases this may take some time. Therefore it may be a good idea to use a common prefix for all *refdb* databases as explained in the section about the **createdb** command.

12.4.12.3. Example

```
refdba: listdb db*
```

This will list all databases with names that start with the string “db”.

12.4.13. liststyle

12.4.13.1. Synopsis

```
liststyle [unix-regexp]
```

12.4.13.2. Description

Lists all available bibliography styles that match *unix-regexp*. *unix-regexp* has to be a valid Unix regular expression. If no argument is given, *all* available styles will be listed. This may or may not be what you want.

Note: Some database engines, like SQLite, do not support Unix-style regular expressions. Use SQL regular expressions instead.

12.4.13.3. Example

```
refdba: liststyle ^J.*
```

This will list all bibliography styles that start with a capital “J”.

12.4.14. listuser

12.4.14.1. Synopsis

```
listuser {-d database} [unix-regexp]
```

12.4.14.2. Description

Lists all available users of the specified database that match *unix-regexp*. *unix-regexp* has to be a valid Unix regular expression. If no argument is given, *all* available users will be listed. This may or may not be what you want.

Note: Some database engines, like SQLite, do not support Unix-style regular expressions. Use SQL regular expressions instead.

12.4.14.3. Example

```
refdba: listuser -d refs ^mo.*
```

This will list all users of the database "refs" whose names start with “mo”.

12.4.15. listword

12.4.15.1. Synopsis

```
listword {unix-regexp}
```

12.4.15.2. Description

Lists all available reserved journal words that match *unix-regexp*. *unix-regexp* has to be a valid Unix regular expression. If no argument is given, *all* available words will be listed. This may or may not be what you want.

Note: Some database engines, like SQLite, do not support Unix-style regular expressions. Use SQL regular expressions instead.

Note: For a brief description of the purpose of reserved words, see the **addword** command.

12.4.15.3. Example

```
refdba: listword ^BIO.*
```

This will list all reserved journal words that start with “BIO”.

12.4.16. scankw

12.4.16.1. Synopsis

```
scankw {-d database} [-h]
```

12.4.16.2. Description

This command schedules a full keyword scan in the database specified with the `-d` option. The abstract field as well as all title fields of all references found in the database are scanned for the presence of all keywords available in the database. If a match is found and the keyword is not yet associated with that reference, the keyword is added to that reference. As the time required to perform this operation increases with both the number of references and the number of keywords, the keyword scan is performed in the background and the command returns immediately on the client side. See the server log for the results.

As this command will cause a huge number of database accesses it is best scheduled to run automatically as a cron job at a time of low use, either nightly or on weekends.

Please note the difference between the full keyword scan and the automatic keyword scan which can be requested by the `refdbd` command line switch `-K` or the corresponding configuration variable `keyword_scan`. The full keyword scan is "retrospective", i.e. it will add keywords that were added later to previously existing references. The automatic keyword scan will only add existing keywords to newly added references, thus causing less impact on the database performance while users are likely to access the database.

12.4.17. set

12.4.17.1. Synopsis

```
set [-h] [varname] [varvalue]
```

12.4.17.2. Description

The **set** command displays or modifies the values of configuration variables.

If you call **set** without any arguments, it will display a list of all configuration variables with their current values.

If you call **set** with one argument, it will display the current value of this particular variable.

If you call **set** with two arguments, it will set the variable (first argument) to the new value (second argument) for the current session. To specify an empty value, use two quotation marks like this: "".

Run **set -h** to see a brief usage message.

Note: For obvious reasons, **set** will never display the current password although you can certainly change the password with this command. To make sure no one else sees the new password that you enter, run the command **set passwd ***. You will then be asked to enter a password which will not be echoed on the screen.

This command is not available in batch mode, use the command line switches instead. In the interactive mode, the changes to the configuration variables are limited to the current session. If you want to change the values permanently, you should rather edit one of the configuration files.

12.4.17.3. Example

```
refdba: set timeout 90
```

This command will set the timeout to 90 seconds for the current session.

12.4.18. verbose

12.4.18.1. Synopsis

verbose

12.4.18.2. Description

Toggles the verbose mode on or off. If the verbose mode is on, the error messages and warnings may be some more comprehensible.

12.4.18.3. Example

```
refdba: verbose
```

Depending on the previous setting, this will toggle the verbose mode on or off.

12.4.19. viewstat

12.4.19.1. Synopsis

```
viewstat
```

12.4.19.2. Description

Shows the version numbers of the libdbi driver used to connect to your database server as well as the version information of that server. It also shows the current values of the variables that can be modified with **confserv**.

12.4.19.3. Example

```
refdba: viewstat
```

This will print some connection statistics and informations on the screen.

Chapter 13. reftdb: the reference management client

13.1. Overview

The following tasks can be performed with reftdb:

- Manually add, edit, and delete reference entries
- Search for and display reference entries
- Search for and display authors and keywords
- Include existing references into personal reference list

13.2. Starting reftdb

To start an interactive session, run reftdb with the command:

```
reftdb [-c pager] [-d database] [-e log-destination] [-F fields] [-g deffile] [-G  
CSS-file] [-h] [-i IP-address] [-l log-level] [-L log-file] [-p port] [-q] [-R pdfroot] [-T  
time] [-u name] [-v] [-V] [-w password] [-y confdir]
```

Unless you specify a password in a configuration file or on the command line, reftdb will ask you for a password before you can proceed. reftdb will then display a prompt and wait for you to enter commands:

```
reftdb:
```

To run a reftdb command from the command line, use:

```
reftdb [-c pager] {-C command-name} {-d database} [-e log-destination] [-F fields] [-g  
deffile] [-G CSS-file] [-h] [-i IP-address] [-l log-level] [-L log-file] [-p port] [-q] [-T  
time] [-u name] [-w password] [-y confdir]
```

This will run the command whose name is specified with the -C switch and return to the command line. In this mode, reftdb can read the input for certain *commands* from stdin. Any additional command-line options that the internal *command* recognizes can be specified on the command line. To find out which options you can use, see the command descriptions below.

Tip: If you prefer to just run the commands like **addref** on the command line, you can (if your shell supports this) create aliases like:

```
alias addref='reftdb -C addref'
```

This works in bash and would go into your `.bashrc`. You can then add references to your database with the following command at the shell prompt:

```
~# addref -d foo mydata.ris
```

Remember that you don't have to specify all command-line options each time if you define the values in `.refdbrc`.

Use the `-d` option to specify the database that you want to work with. In an interactive session you can also set and change the default database with the **selectdb** command.

The `-F` option specifies the default fields that are to be displayed in a **getref** query.

The `-g` option can be used to add some default fields to all references that are added or updated. The argument *deffile* is the filename of a RIS file containing these additional fields. `refdbc` first tries the filename as is, so it should be a valid relative or absolute path. If the file is not found, `refdb` looks for the file in `$HOME`. The command aborts if the file cannot be found.

Use the `-G` option to specify the URL of a Cascading Style Sheets (CSS) file. This file will be used to customize the HTML output of the **getref** command. The URL can be either a local path (e.g. `refdb.css`, `/home/myname/custom.css`) or the web address of a file on a web server (e.g. `http://www.mycomp.com/refdb.css`).

The `-R` option specifies the root path of your collection of electronic offprints. See the information about `pdfroot` for further details.

The purpose of all other command-line switches is explained in Common command-line options.

Note: If you want to run `refdbc` unattended in batch-mode from a script, you should make sure that you do not specify a pager or a pipe command for an interactive program for the output. This would block the script until you hit a button. To this end you can add the `-c stdout` option to the command line to disable any pipe or pager commands that you may have in the `.refdbrc` file.

You may also want to redirect the `stderr` output to a file for logging purposes.

The exit code is 0 if all went fine. It will be 1 if the command (when run in batch mode) or the last command (when run in interactive mode) returned an error, or if there was a general error condition during startup like a lack of available memory.

13.3. The `refdbc` configuration file variables

`refdbc` reads the file `refdbrc` at startup and uses the variable/value pairs to initialize the application. You may override these settings by providing the appropriate command line options.

Table 13-1. `refdbrc`

Variable	Default	Comment
cssurl	(none)	The URL of a Cascading Style Sheet (CSS) file. This file, if specified, is used to customize the visual appearance of the HTML output of the getref command.
defaultdb	(none)	The default database. refdbc will try to use this database unless you select a different one with the selectdb command.
defaulttris	(none)	The path of a RIS file with entries that should be added to all new or updated references. This is typically used to set some default value for the RP field or to specify additional keywords.
fields	(none)	A list of additional fields which should be displayed by default in the reference output. The list is a simple concatenation of the field names. Possible fields are N1, N2, NX, AB, AD, RP, SN, PB, CY, UR, U1 through U5, M1 through M3. Use the string "ALL" to request all available fields.
fromencoding	ISO-8859-1	The default encoding of RIS input data. You can use any encoding that your local libiconv implementation supports.
logdest	file	Where the log output should be written to. Use either stderr, syslog, or file. For the latter to work, the logfile variable must be set appropriately
logfile	/var/log/refdbc.log	The full path of a custom log file.
loglevel	info	Set the level of log information that you would receive. Possible values, in order of increasing verbosity, are: emerg, alert, crit, err, warning, notice, info, debug

Variable	Default	Comment
pager	stdout	The command line of a pager that accepts the output of refdb on stdin to allow scrolling and other nifty things. “stdout” sends the data to stdout.
passwd	*	The password which is used for authentication with the database server. It is potentially evil to store unencrypted passwords in disk files. At least make sure that the configuration file is not readable for anyone else. The default setting causes refdbbc to ask for your password interactively.
pdfroot	(none)	This value will be used as the root of the paths to PDF or Postscript offprints that can be specified with the AV field in a RIS dataset. The path should not rely on shell expansion, e.g. use /home/me/literature/ instead of ~/literature/. The pdfroot allows you to shorten the paths that you enter for each dataset and to maintain a certain portability if you have to move the offprints to a different directory or want to access them remotely. The html output routine will concatenate the relative path of each dataset with the pdfroot to construct the link to the offprint. Instead of a local path name you can specify an URL starting with http:// or ftp:// if your offprints are accessible through a web server or ftp server.
port	9734	The port on which refdbd listens. Change this for all clients and the server if this value interferes with another program using this port.

Variable	Default	Comment
serverip	127.0.0.1	The IP address or hostname of the machine where refdbd runs. Use the default (localhost) address if the clients and refdbd run on the same machine.
timeout	180	The timeout in seconds. After this time has elapsed, a stalled connection is taken down. Increase this value if you encounter frequent timeout errors due to high network traffic or refdbd overload.
toencoding	(none)	The default encoding of output data. You can use any encoding that your local libiconv implementation supports. If this value is not set, the encoding of the database will be used without conversion.
username	login name	The username which is used for authentication with the database server. This may be different from the login name of the user.
verbose	f	Set this to t if you prefer verbose error messages.

13.4. Commands

The following sections explain all commands available in refdbc.

13.4.1. addlink

13.4.1.1. Synopsis

addlink [-d *database*] [-h] [-c *command*] [-o *outfile*] [-O *outfile*] {note-specifier} {link-target...}

13.4.1.2. Description

The **addlink** command links an extended note to one or more link targets.

You have to specify exactly one note on the command line, either by using the `:NID:` field selector to specify the note ID, or by using the `:NCK:` field selector to specify the note key. Then you need at least one link target. This can be one of `:ID:` (reference by ID), `:CK:` (reference by citation key), `:AU:` (author by name), `:KW:` (keyword by name), or one of `:JF:`, `:JO:`, `:J1:`, `:J2:` (periodical by full name, abbreviated name, or user abbreviations 1 and 2).

13.4.1.3. Example

```
refdbc: addlink :NID:=12 :CK:=Miller1999 :KW:=biochemistry
```

This command will link the note carrying the ID 12 with a reference specified by its citation key "Miller1999" and with the keyword "biochemistry".

13.4.2. addnote

13.4.2.1. Synopsis

```
addnote [-d database] [-E encoding] [-h] [-c command | -o outfile | -O outfile] [file...]
```

13.4.2.2. Description

Adds the extended notes in *file* to the current database. You can specify several files in one run. Any ID fields in the notes are ignored.

The `-c` switch allows to specify a **command**. *refdbc* will open a pipe to the first program in the command and send the output to this program's stdin. The command may be any valid command that you can run in your shell, so further plumbing is perfectly legal.

Use the `-d` option to specify the database that you want to work with if it is different from the currently selected database.

Select an input character encoding with the `-E` option if it is different from the default UTF-8.

The `-h` option displays a short command syntax and description, then returns to the command prompt.

The `-o` and `-O` switches allow to redirect the output to *outfile* instead of the default screen display. The two options differ in the way they handle an existing *outfile*. `-o` will replace the existing file, while `-O` will append to the existing file. If *outfile* cannot be opened with the proper permissions, the output is sent to stdout instead.

All other command-line arguments will be interpreted as filenames to read references from. If no filenames are specified, the data will be read from stdin.

13.4.2.3. Example

```
refdbc: addnote foo.xml
```

This command will add the extended notes in *foo.ris* to the currently selected database. If the notes do not specify a date, *refdbd* will insert a timestamp automatically.

13.4.3. *addref*

13.4.3.1. Synopsis

```
addref [-d database] [-E encoding] [-g deffile] [-h] [-c command | -o outfile | -O outfile] [-t type] [-U username] [file...]
```

13.4.3.2. Description

Adds the references in *file* to the current database. You can specify several files in one run. Any ID fields in the references are ignored unless you specify the *-k* option.

The *-c* switch allows to specify a **command**. *refdbc* will open a pipe to the first program in the command and send the output to this program's stdin. The command may be any valid command that you can run in your shell, so further plumbing is perfectly legal. This feature may e.g. be used to filter the output with *grep* for the error messages, dropping all success messages. This is of course not intended to make your world look grey and dull, but to make it easier to spot the (hopefully zero or few) error messages inbetween all those success messages.

Use the *-d* option to specify the database that you want to work with.

Select an input character encoding with the *-E* option if it is different from the default UTF-8. RIS datasets can use any encoding that your local *libiconv* supports (see **man iconv_open** for a list of available encodings), except UTF-16 and UTF-32. RISX datasets carry the encoding in the processing instructions, therefore this option is ignored.

You can use two different input file formats with this command. The default format is the tagged RIS format. Use *-t risx* to use XML files according to the RISX DTD as input data.

The *-g* option can be used in conjunction with RIS data to add some default fields to all references that are added with this command. The argument *deffile* is the filename of a RIS file containing these additional fields. *refdbc* first tries the filename as is, so it should be a valid relative or absolute path. If the file is not found, *refdb* looks for the file in *\$HOME*. The command aborts if the file cannot be found.

Keep in mind that this default file, just like any other RIS file, has to start with a newline and needs at least the TY and ER fields.

The following example RIS file would switch the reprint status of all added references to “ON REQUEST” as of Jan 1, 2001 and let them have the type “journal” (please note the mandatory empty first line created by a LF):

```
TY   - JOUR
RP   - ON REQUEST(01/01/01)
ER   -
```

The `-h` option displays a short command syntax and description, then returns to the command prompt.

The `-o` and `-O` switches allow to redirect the output to `outfile` instead of the default screen display. The two options differ in the way they handle an existing `outfile`. `-o` will replace the existing file, while `-O` will append to the existing file. If `outfile` cannot be opened with the proper permissions, the output is sent to `stdout` instead.

The `-U` switch allows to provide a different username than the current user, so e.g. some technician or administrative staff can add references in behalf of a researcher.

All other command-line arguments will be interpreted as filenames to read references from. If no filenames are specified, the data will be read from `stdin`.

13.4.3.3. Example

```
refdbc: addref -U doe -g .refdbdefault.ris -E ISO-8859-1 foo.ris
```

```
$ refdbc -C addref -U doe -g .refdbdefault.ris -d db1 < foo.ris
```

These commands will add the references in `foo.ris`. The references will be associated with the user “doe”. Every reference will use the specified values in `.refdbdefault.ris` in the appropriate fields. In the first (interactive) command, the active database will be used, and the encoding is set to ISO-8859-1, aka Latin-1. In the second (non-interactive) command, the database has to be specified explicitly with the `-d` option, and the default encoding (UTF-8) is assumed.

13.4.4. deletelink

13.4.4.1. Synopsis

```
deletelink [-d database] [-h] [-c command] [-o outfile] [-O  
outfile] {note-specifier} {link-target...}
```

13.4.4.2. Description

The **deletelink** command removes links from an extended note to one or more link targets.

You have to specify exactly one note on the command line, either by using the :NID: field selector to specify the note ID, or by using the :NCK: field selector to specify the note key. Then you need at least one link target. This can be one of :ID: (reference by ID), :CK: (reference by citation key), :AU: (author by name), :KW: (keyword by name), or one of :JF:, :JO:, :J1:, :J2: (periodical by full name, abbreviated name, or user abbreviations 1 and 2).

13.4.4.3. Example

```
refdbc: deletelink :NID:=12 :CK:=Miller1999 :KW:=biochemistry
```

This command will delete the links from the note carrying the ID 12 to a reference specified by its citation key "Miller1999" and to the keyword "biochemistry".

13.4.5. deletenote

13.4.5.1. Synopsis

```
deletenote [-d database] [-h] [-c command | -o outfile | -O outfile] { ID | -f infile }...
```

13.4.5.2. Description

Deletes the extended note with the identifier *ID* from the current database. Several extended notes may be specified in a single call of this command. Notes with consecutive ID values may be specified as ranges.

Alternatively you can specify a file with the **-f** switch which contains a list of IDs in the RIS format. This can e.g. be a previously saved search with the **-o** and **-s ID** options of **getnote**. The **-f** switch can also be combined with IDs specified as arguments. If no IDs are specified, the ID list is read from stdin. If you want to read from stdin *in addition to* other IDs, you must use the **-f stdin** option.

The **-c** switch allows to specify a shell **command**. *refdbc* will open a pipe to the first program in the command and send the output to this program's stdin. The command may be any valid command that you can run in your shell, so further plumbing is perfectly legal. This feature may e.g. be used to filter the output with **grep**.

Use the **-d** option to specify the database that you want to work with.

The **-h** option displays a short command syntax and description, then returns to the command prompt.

The **-o** and **-O** switches allow to redirect the output to *outfile* instead of the default screen display. The two options differ in the way they handle an existing *outfile*. **-o** will replace the existing file,

while `-O` will append to the existing file. If `outfile` cannot be opened with the proper permissions, the output is sent to `stdout` instead.

Caution

It is not possible to delete a note if it belongs to a different user.

13.4.5.3. Example

```
refdbc: deletenote 3 5-10 26
```

This command will delete the extended notes with the ID values 3, 5 through 10, and 26.

13.4.6. deleteref

13.4.6.1. Synopsis

```
deleteref [-d database] [-h] [-c command] [-o outfile] [-O outfile] [ID] [-f infile]{...}
```

13.4.6.2. Description

Deletes the reference with the identifier *ID* from the current database. Several references may be specified in a single call of this command. References with consecutive ID values may be specified as ranges.

Alternatively you can specify a file with the `-f` switch which contains a list of IDs in the RIS format. This can e.g. be a previously saved search with the `-o` and `-s ID` options of **getref**. The `-f` switch can also be combined with IDs specified as arguments. If no IDs are specified, the ID list is read from `stdin`. If you want to read an ID list from `stdin` *in addition to* IDs passed as arguments, you have to use the `-f stdin` option.

The `-c` switch allows to specify a shell **command**. `refdbc` will open a pipe to the first program in the command and send the output to this program's `stdin`. The command may be any valid command that you can run in your shell, so further plumbing is perfectly legal. This feature may e.g. be used to filter the output with `grep`.

Use the `-d` option to specify the database that you want to work with.

The `-h` option displays a short command syntax and description, then returns to the command prompt.

The `-o` and `-O` switches allow to redirect the output to `outfile` instead of the default screen display. The two options differ in the way they handle an existing `outfile`. `-o` will replace the existing file,

while `-O` will append to the existing file. If `outfile` cannot be opened with the proper permissions, the output is sent to `stdout` instead.

Caution

It is not possible to delete a reference if it belongs to the personal reference list of more than one user.

If you're the only user of this reference and go ahead and delete it, all data saved in the specified references will be gone, so be careful with this command. Make sure you understand the difference between the **deleteref** command and the **dumpref** command. The former deletes the data, the latter deletes only your personal data associated with the specified references (the notes, availability, and reprint data) and removes your association with this reference. In other words, **deleteref** removes the reference from the database, whereas **dumpref** removes the reference from your personal reference list, leaving the remaining data for the other users of the database.

13.4.6.3. Example

```
refdbc: deleteref 3 5-10 26
```

This command will delete the references with the ID values 3, 5 through 10, and 26.

13.4.7. dumpref

13.4.7.1. Synopsis

```
dumpref [-d database] [-h] [-c command | -o outfile | -O outfile] {ID | -f file}...
```

13.4.7.2. Description

Removes references from your personal reference list in the current database.

The `-c` switch allows to specify a **command**. `refdbc` will open a pipe to the first program in the command and send the output to this program's `stdin`. The command may be any valid command that you can run in your shell, so further plumbing is perfectly legal. This feature may e.g. be used to filter the output with `grep`.

Use the `-d` option to specify the database that you want to work with.

The `-h` option displays a short command syntax and description, then returns to the command prompt.

The `-o` and `-O` switches allow to redirect the output to `outfile` instead of the default screen display. The two options differ in the way they handle an existing `outfile`. `-o` will replace the existing file, while `-O` will append to the existing file. If `outfile` cannot be opened with the proper permissions, the output is sent to `stdout` instead.

The references can be specified on the command line or they can be read from a file which is specified with the `-f` option. Both methods can be combined. The file has to be in RIS format and can e.g. be a previously saved search with the `-o` and `-s ID` options of **getref**. If no IDs are specified as arguments, the ID list is read from `stdin`. If you want to read from `stdin` *in addition to* IDs specified as arguments, you have to use the `-f stdin` option.

13.4.7.3. Example

```
refdbc: dumpref -f foo.ris 3 5-10 26
```

This command will remove the references 3, 5 through 10, and 26 as well as those listed in the file `foo.ris` from your personal reference list.

13.4.8. getau, geted, getas

13.4.8.1. Synopsis

```
getau [-d database] [-h] [-c command | -o outfile | -O outfile] {unix-regexp}
```

```
geted [-d database] [-h] [-c command | -o outfile | -O outfile] {unix-regexp}
```

```
getas [-d database] [-h] [-c command | -o outfile | -O outfile] {unix-regexp}
```

13.4.8.2. Description

Retrieve all author names that match the regular expression `unix-regexp` in the current database. If no `regexp` argument is given, *all* author names will be listed, which may or may not be what you want.

getau retrieves the *primary* authors which is the most common case if you want to locate a publication.

geted and **getas** retrieve book or periodical editors and series authors, respectively.

Note: Some database engines, like SQLite, do not support Unix-style regular expressions. Use SQL regular expressions instead.

The `-c` switch allows to specify a **command**. `refdbc` will open a pipe to the first program in the command and send the output to this program's stdin. The command may be any valid command that you can run in your shell, so further plumbing is perfectly legal.

The `-h` option displays a short command syntax and description, then returns to the command prompt.

The `-o` and `-O` switches allow to redirect the output to `outfile` instead of the default screen display. The two options differ in the way they handle an existing `outfile`. `-o` will replace the existing file, while `-O` will append to the existing file. If `outfile` cannot be opened with the proper permissions, the output is sent to stdout instead.

13.4.8.3. Example

```
refdbc: getau -o authors.txt '^Simpson'
```

This command will print a list of all authors starting with “Simpson”.

13.4.9. `getjo`, `getjf`, `getj1`, `getj2`

13.4.9.1. Synopsis

```
getjo [-a] [-d database] [-h] [-c command | -o outfile | -O outfile] {unix-regexp}
```

```
getjf [-a] [-d database] [-h] [-c command | -o outfile | -O outfile] {unix-regexp}
```

```
getj1 [-a] [-d database] [-h] [-c command | -o outfile | -O outfile] {unix-regexp}
```

```
getj2 [-a] [-d database] [-h] [-c command | -o outfile | -O outfile] {unix-regexp}
```

13.4.9.2. Description

Retrieve all journal names that match the regular expression *unix-regexp* in the current database. The *unix-regexp* will be matched to the journal abbreviation, the full name, the custom abbreviation 1, and the custom abbreviation 2, respectively. If no regexp argument is given, *all* available journal names will be listed.

Note: Some database engines, like SQLite, do not support Unix-style regular expressions. Use SQL regular expressions instead.

If the `-a` option is present, all synonymous journal names will be returned (i.e. full name, abbreviation, custom abbreviation 1, and custom abbreviation 2). If the option is absent, only the name that you search for will be returned (e.g. only the full name in the case of **getjf**).

The `-c` switch allows to specify a **command**. *refdbc* will open a pipe to the first program in the command and send the output to this program's stdin. The command may be any valid command that you can run in your shell, so further plumbing is perfectly legal.

Use the `-d` option to specify the database that you want to work with.

The `-h` option displays a short command syntax and description, then returns to the command prompt.

The `-o` and `-O` switches allow to redirect the output to `outfile` instead of the default screen display. The two options differ in the way they handle an existing `outfile`. `-o` will replace the existing file, while `-O` will append to the existing file. If `outfile` cannot be opened with the proper permissions, the output is sent to stdout instead.

13.4.9.3. Example

```
refdbc: getjo -a -o journals.txt '^J'
```

This command will list all synonyms of the journals whose abbreviations start with a capital J. The output will be redirected into the file `journals.txt`.

13.4.10. getkw

13.4.10.1. Synopsis

```
getkw [-d database] [-h] [-c command | -o outfile | -O outfile] {unix-regexp}
```

13.4.10.2. Description

Retrieve all keywords that match the regular expression *unix-regexp* in the current database. If no *regexp* argument is specified, all keywords in the database will be listed.

Note: Some database engines, like SQLite, do not support Unix-style regular expressions. Use SQL regular expressions instead.

The `-c` switch allows to specify a **command**. *refdbc* will open a pipe to the first program in the command and send the output to this program's stdin. The command may be any valid command that you can run in your shell, so further plumbing is perfectly legal.

The `-h` option displays a short command syntax and description, then returns to the command prompt.

The `-o` and `-O` switches allow to redirect the output to `outfile` instead of the default screen display.

The two options differ in the way they handle an existing `outfile`. `-o` will replace the existing file, while `-O` will append to the existing file. If `outfile` cannot be opened with the proper permissions, the output is sent to `stdout` instead.

13.4.10.3. Example

```
refdbc: getkw -o keywords.txt '^An.*l$'
```

This command will request a list of all keywords that start with “An” and end with the letter “l”, like “Animal”, and write the result to the file `keywords.txt`.

13.4.11. `getnote`

13.4.11.1. Synopsis

```
getnote [-d database] [-E encoding] [-h] [-c command | -o outfile | -O outfile] [-P] [-s format-string] [-S sort-string] [-t output-format] { search-string | -f file }
```

13.4.11.2. Description

Displays all extended notes which match the *search-string* in the current database. Refer to the section The query language for a description of the syntax of a search string.

The `-c` switch allows to specify a shell **command**. `refdbc` will open a pipe to the first program in the command and send the output to this program’s `stdin`. The command may be any valid command that you can run in your shell, so further plumbing is perfectly legal. This command is handy if you want to search potentially long fields like the content for certain strings. Searching all abstracts of a database with a normal query is slow. It is usually faster to narrow down the search using other fields as far as possible without including the content field and then use `grep` to find what you want.

Use the `-d` option to specify the database that you want to work with.

The retrieved data will use the character encoding of the database unless you request a different encoding with the `-E` option. All encodings supported by your local `libiconv` installation may be specified here.

The `-h` option displays a short command syntax and description, then returns to the command prompt.

The `-o` and `-O` switches allow to redirect the output to `outfile` instead of the default screen display.

The two options differ in the way they handle an existing `outfile`. `-o` will replace the existing file, while `-O` will append to the existing file. If `outfile` cannot be opened with the proper permissions, the output is sent to `stdout` instead.

Caution

Depending on your query, the `getnote` command can generate an enormous amount of output. If you view the output with a pager, the client-server communication will stall as soon as the pager accepts no new data. If the connection times out, your query results will be incomplete. It is strongly recommended to redirect all queries which return a lot of notes (rule of thumb: more than 100 for screen output, more than 50 for other output) to a file or to a pipe that can handle the amount of data.

The `-S` switch is used to sort the output. Currently you can sort only by *ID* (the default) or by *PY* (publication year).

The `-P` switch limits the search to the notes which were added by the current user. If this switch is absent, the whole database will be searched.

The `-t` switch determines the type of output. The default value for *output-format* is *scrn* (screen output), other possible values are *xnote* (XML according to the *xnote* DTD), *html* (HTML), and *xhtml*.

The `-f` switch reads the search string from *file* instead of from the command line, thus allowing to save searches which will be run repeatedly. If no search string is specified, the string will be read from *stdin*.

The syntax of the queries is described in the section query language.

13.4.11.3. Example

```
refdbc: getnote -t xnote :CK:=Miller1999
```

This command retrieves notes which are attached to the reference with the citation key "Miller1999" and displays them in the *xnote* format.

13.4.12. `getref`

13.4.12.1. Synopsis

```
getref [-d database] [-E encoding] [-h] [-c command | -o outfile | -O outfile] [-P] [-s  
format-string] [-S sort-string] [-t output-format] {search-string | -f file}
```

13.4.12.2. Description

Displays all datasets which match the *search-string* in the current database. Refer to the section The query language for a description of the syntax of a search string.

The `-c` switch allows to specify a shell **command**. `refdb` will open a pipe to the first program in the command and send the output to this program's stdin. The command may be any valid command that you can run in your shell, so further plumbing is perfectly legal. This command is handy if you want to search potentially long fields like the abstracts for certain strings. Searching all abstracts of a database with a normal query is slow. It is usually faster to narrow down the search using other fields as far as possible without including the `N2` field and then use `grep` to find what you want.

Use the `-d` option to specify the database that you want to work with.

The retrieved data will use the character encoding of the database unless you request a different encoding with the `-E` option. All encodings supported by your local `libiconv` installation may be specified here. See **man iconv_open** for a list of available encodings.

The `-h` option displays a short command syntax and description, then returns to the command prompt.

The `-o` and `-O` switches allow to redirect the output to `outfile` instead of the default screen display. The two options differ in the way they handle an existing `outfile`. `-o` will replace the existing file, while `-O` will append to the existing file. If `outfile` cannot be opened with the proper permissions, the output is sent to stdout instead.

Caution

Depending on your query, the `getref` command can generate an enormous amount of output. If you view the output with a pager, the client-server communication will stall as soon as the pager accepts no new data. If the connection times out, your query results will be incomplete. It is strongly recommended to redirect all queries which return a lot of references (rule of thumb: more than 100 for screen output, more than 50 for other output) to a file or to a pipe that can handle the amount of data.

Except for RIS and `risx` output which always display the full dataset, the `-s` switch allows to specify additional fields (`N1`, `N2/AB`, `NX`, `RP`, `SN`, `AD`, `CY`, `PB`, `UR`, `U1` through `U5`, `M1` through `M3`) that are not displayed by default. Use "ALL" as an argument to display all available fields. If several fields are specified, the argument has to be enclosed by single quotation marks. If applied to RIS output, you can specify `ID` as *format-string* to get only a list of ID values in RIS format for all references that match the search. This is a convenient way to generate ID lists for later operations like **deleteref**.

The `-S` switch is used to sort the output. Currently you can sort only by `ID` (the default) or by `PY` (publication year).

The `-P` switch limits the search to the files which are in the current user's personal reference list. If this switch is absent, the whole database will be searched.

The `-t` switch determines the type of output. The default value for *output-format* is `scrn` (screen output), other possible values are `db3l` (DocBook SGML V. 3.1), `db3lx` (DocBook XML), `ris` (RIS as of Reference Manager 8.01), `risx` (XML according to the `risx` DTD), `html` (HTML), `xhtml` (XHTML), and `bibtex` (BibTeX).

The `-f` switch reads the search string from `file` instead of from the command line, thus allowing to save searches which will be run repeatedly. If no search string is specified, the string is read from stdin.

The syntax of the queries is described in the section query language.

13.4.12.3. Example

```
refdbc: getref -t ris -o temp.sgml -E ISO-8859-15 ":AU:=' & ^Doe ^Jones' AND :KW:=circular\
```

This command retrieves articles with both an author starting with “Doe” and an author starting with “Jones” that have the keyword “circular dichroism”. The output will be saved in RIS format to the file `temp.sgml` using the character encoding ISO-8859-15.

13.4.13. help, ?

13.4.13.1. Synopsis

help

?

13.4.13.2. Description

Displays a brief summary of the available commands.

Note: This command is not available in the batch mode (use the `-h` option instead to review the command line usage).

13.4.13.3. Example

```
refdbc: help
```

This will list the available commands.

13.4.14. listdb

13.4.14.1. Synopsis

listdb [-h] [*SQL-regexp*]

13.4.14.2. Description

Lists all available databases if no argument is specified. If *SQL-regexp* is specified, only the databases matching this expression will be listed. *SQL-regexp* has to be a valid SQL regular expression.

The `-h` option displays a short command syntax and description, then returns to the command prompt.

13.4.14.3. Example

```
refdbc: listdb db%
```

This command will list all available databases that start with “db”.

13.4.15. liststyle

13.4.15.1. Synopsis

liststyle {*unix-regexp*}

13.4.15.2. Description

Lists all available bibliography styles that match *unix-regexp*. *unix-regexp* has to be a valid Unix regular expression. If no argument is specified, all available styles will be listed.

Note: Some database engines, like SQLite, do not support Unix-style regular expressions. Use SQL regular expressions instead.

13.4.15.3. Example

```
refdba: liststyle ^J.*
```


This will list all bibliography styles that start with a capital “J”.

13.4.16. pickref

13.4.16.1. Synopsis

```
pickref [-d database] [-h] [-c command | -o outfile | -O outfile] { ID | -f file }...
```

13.4.16.2. Description

Adds references to your personal reference list in the current database.

The `-c` switch allows to specify a **command**. `refdbc` will open a pipe to the first program in the command and send the output to this program’s stdin. The command may be any valid command that you can run in your shell, so further plumbing is perfectly legal. This feature may e.g. be used to filter the output with `grep`.

Use the `-d` option to specify the database that you want to work with.

The `-h` option displays a short command syntax and description, then returns to the command prompt.

The `-o` and `-O` switches allow to redirect the output to `outfile` instead of the default screen display. The two options differ in the way they handle an existing `outfile`. `-o` will replace the existing file, while `-O` will append to the existing file. If `outfile` cannot be opened with the proper permissions, the output is sent to stdout instead.

The references can be specified on the command line or they can be read from a file which is specified with the `-f` option. Both methods can be combined. The file has to be in RIS format and can e.g. be a previously saved search with the `-o` and `-s ID` options of **getref**. If no IDs are specified, the ID list in RIS format is read from stdin.

13.4.16.3. Example

```
refdbc: pickref -f foo.ris 3 5-10 26
```

This command will add the references 3, 5 through 10, and 26 as well as those listed in the file `foo.ris` to your personal reference list.

13.4.17. selectdb

13.4.17.1. Synopsis

```
selectdb [-h] {database}
```

13.4.17.2. Description

Select *database* as the current database. This current database will be used in all queries unless you specify a different database with the `-d` option of the query commands.

Note: This command is not available in the batch mode. Use the `-d` command line option instead.

The `-h` option displays a short command syntax and description, then returns to the command prompt.

13.4.17.3. Example

```
refdbc: selectdb db1
```

This command will make the database `db1` the active database. All further queries and operations will affect this database.

13.4.18. set

13.4.18.1. Synopsis

```
set [-h] [varname] [varvalue]
```

13.4.18.2. Description

The **set** command displays or modifies the values of configuration variables.

If you call **set** without any arguments, it will display a list of all configuration variables with their current values.

If you call **set** with one argument, it will display the value of this particular variable.

If you call **set** with two arguments, it will set the variable (first argument) to the new value (second argument). To specify an empty value, use two quotation marks like this: "".

Run **set -h** to see a brief usage message.

Note: For obvious reasons, **set** will never display the current password although you can certainly change the password with this command. To make sure no one else sees the new password that you enter, run the command **set passwd ***. You will then be asked to enter a password which will not be echoed on the screen.

This command is not available in batch mode, use the command line switches instead. In the interactive mode, the changes to the configuration variables are limited to the current session. If you want to change the values permanently, you should rather edit one of the configuration files.

13.4.18.3. Example

```
refdbc: set timeout 90
```

This command will set the timeout to 90 seconds for the current session.

13.4.19. updatenote

13.4.19.1. Synopsis

```
updatenote [-d database] [-E encoding] [-h] [-c command | -o outfile | -O outfile] [file...]
```

13.4.19.2. Description

Updates the extended notes in *file* in the current database.

This command is essentially the same as **addnote**, but it uses the `citekey` or `id` attributes (in this order) to update an existing note in the database. If the specified note does not exist in the database, a new one will be created.

The command line options are the same as described for the **addnote** command.

13.4.19.3. Example

```
refdbc: updatenote foo.xml
```

This command will update the extended notes in `foo.ris` in the currently selected database. If the notes do not specify a date, **refdbd** will insert a timestamp automatically.

13.4.20. updateref

13.4.20.1. Synopsis

```
updateref [-d database] [-E encoding] [-g deffile] [-h] [-c command | -o outfile | -O  
outfile] [-t type] [-P] [-U username] [file]
```

13.4.20.2. Description

Updates the references in RIS format in *file* in the current database.

This command is essentially the same as **addref**, but it uses the *ID* fields in the input data to update existing references with the same ID. If the ID of a reference is not existent in the database, a new entry is created, ignoring the ID specified in the RIS or risx file. Currently **refdb** does not check whether the new dataset has any similarity with the old one having the same ID. If you tell **refdb** to update a reference, it uses whatever you send to this end.

For the **-c**, **-E**, **-g**, **-h**, **-o**, **-O**, **-U**, and **-f** options, please refer to the description of the **addref** command.

Use the **-P** switch to update only the personal information for this reference, i.e. the N1 (notes), RP (reprint status), and AV (availability) fields. This will automatically add the reference to your personal reference list. All other fields will be ignored. Combine this option with the **-g** option e.g. to quickly change the reprint status of existing references to “IN FILE” from “NOT IN FILE” or from “ON REQUEST”.

13.4.20.3. Example

```
refdbc: updateref -P foo.ris
```

This command will update the references in *foo.ris* in the previously selected active database. Only the personal information (AV, N1, RP) will be added or modified for the current user.

13.4.21. verbose

13.4.21.1. Synopsis

```
verbose [-h]
```

13.4.21.2. Description

Toggles the verbose mode on or off. If the verbose mode is on, the error messages and warnings may be some more comprehensible.

The `-h` option displays a short command syntax and description, then returns to the command prompt.

13.4.21.3. Example

```
refdbc: verbose
```

Depending on the previous value, this command will either turn the verbose mode on or off.

13.4.22. **whichdb**

13.4.22.1. Synopsis

```
whichdb [-h]
```

13.4.22.2. Description

Displays a plethora of information about the currently selected database.

The `-h` option displays a short command syntax and description, then returns to the command prompt.

13.4.22.3. Example

```
refdbc: whichdb
```

This will print the information about the active database. Refer to the **selectdb** command for information how to change the active database. The **whichdb** output looks like this:

```
Current database: alltypes
Number of references: 45
Highest reference ID: 45
Number of notes: 2
Highest note ID: 2
Encoding: ISO-8859-1
Database type: risx
Server type: pgsq1
Created: 2003-12-24 22:27:43 UTC
Using refdb version: 0.9.4-pre2
Last modified: 2003-12-24 22:29:05 UTC
```

13.5. Reference data output formats

refdbd implements several standard backends for data output. You can select them with the `-t` option of the **getref** command in `refdbc`. Remember that all backends just send data. It is your decision what you want to do with the information. You can either view the data on the screen or pipe them to another application or write them to a disk file.

13.5.1. `scrn`

The screen backend provides a basic data output for viewing in a terminal, preferably through a pager. By default, the reference ID, the publication year, the authors, the title, and the source information are displayed. You can use the `-s` option to additionally display the abstract (AB or N2), the notes (N1), the reprint info (RP), the address (AD), the publisher (PB), the city (CY), the URL (UR), and the user (U1 through U5) and misc (M1 through M3) fields. `-s ALL` will display all available fields.

13.5.2. `html`

The html backend works just like the `scrn` backend, but encodes this information in a HTML text. This comes in handy if you would like to view the results of your queries in a web browser rather than in a terminal window. This is also the easiest way to obtain fairly nice-looking printed output of your reference data. You simply use the `-o` switch to write the results of your queries to a file, reusing the same filename for each query. After each query you just have to hit the reload button of your browser to view the results of the most recent query.

The visual appearance of the generated HTML files can be customized using CSS files. The URL of the CSS file must be specified with the `refdbc -G` command-line option or by setting the configuration variable `cssurl`. The global configuration file `/usr/local/etc/refdb/refdbcrc` is preconfigured with a `cssurl` entry pointing to the default CSS file installed in `/usr/local/share/refdb/css`. If you want a different appearance, it might be prudent to create a copy of this CSS file and customize it as you see fit. The following element classes can be customized. In most cases, the class name reflects the database field to be formatted:

- H1.h1
- H2.id
- P.title
- P.authors
- P.abstract
- P.note
- P.address
- P.city
- P.publisher

- P.m1, P.m2, P.m3
- P.u1, P.u2, P.u3, P.u4, P.u5
- P.url
- EM.periodical
- EM.volume
- EM.issue
- EM.page

13.5.3. xhtml

This is a variant of the html output which creates valid XML output according to the XHTML V1.0 Transitional DTD (<http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd>).

13.5.4. ris

This is identical to the input format. Use it to export references to other reference management systems.

13.5.5. risx

Retrieves the data as an XML document using the risx DTD. Use it to edit references or to create backups of your databases.

13.5.6. bibtex

This backend provides output formatted for use as a bibtex reference database. This can be used with the tex and bibtex applications to create bibliographies for documents written with Donald Knuth's famous TeX (<http://www.tug.org>) typesetting system. The `-s` option cannot be used with this backend and will be ignored.

13.5.7. db31

The DocBook SGML backend formats the query result as a `bibliography` element in a SGML document using the DocBook DTD. `refdb` outputs an appropriate doctype string at the beginning of the data. The string is commented out so the contents can be directly inserted into a larger document by some processing application. If you need the data as a standalone document, simply use a script to uncomment the first line. The `-s` option cannot be used with this backend and will be ignored. This option is called "db31" to distinguish it from any later additions that may be necessary due to possible

incompatibilities in new DocBook versions. It just means that the SGML that this backend creates should work with DocBook V.3.1 or later unless a new backend will be added.

13.5.8. db31x

The output is essentially the same as with the preceeding backend but you'll get a DocBook XML document instead.

13.5.9. teix

The TEI XML backend formats the query results as a TEI `listBibl` element. `refdb` outputs an appropriate processing instruction and doctype string at the beginning of the data. The string is commented out so the contents can be directly inserted into a larger document by some processing application. If you need the data as a standalone document, simply use a script to uncomment the first line. The `-s` option cannot be used with this backend and will be ignored.

13.6. Extended notes output formats

`refdb` implements a few backends to output extended notes. You can select them with the `-t` option of the `getnote` command.

13.6.1. scrn

The screen backend provides the output most suitable for viewing notes in a terminal. Send the output through a pager for best results. Use the `-s NL` or `-s ALL` options to display all objects the note is linked to.

13.6.2. html

The information returned by this backend is encoded as a HTML document. Use this format to print nicely formatted notes from your web browser. Otherwise the same applies as said for the `scrn` backend. See above for some hints about formatting the output with a CSS file.

13.6.3. xhtml

The output is the same as for the `html` backend but the output is formatted as an XHTML document according to the XHTML V1.0 Transitional DTD (<http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd>).

13.6.4. xnote

This backend encodes the output in xnote (<http://refdb.sourceforge.net/dtd/xnote/xnote.dtd>) documents, the same as the input format for extended notes.

13.7. The query language

The **getref** command is probably the most heavily used command. You use it to retrieve the references that you collected and saved in the database. To find a certain article or several related articles, all you have to do is to express your query in a language that refdb understands. The first section describes how to formulate search strings for your queries.

The **getnote** command used to locate extended notes is very similar. The specifics of this command will be described in the subsequent section.

13.7.1. The reference query language

The syntax for the search string follows these rules:

- You can search for any fields in the refdb database. As a matter of fact, you have to specify at least one field for your query. Something like **getref *** will not work, but you may use **getref ":ID:>0"** instead to list all entries in the database (beware, this may be a *lot*).
- Every search item has the following general form:

```
:XY:[|=|!|=|!~|<|>]string
```

Warning

The current implementation of refdb is very picky about spaces. Please make sure that you do not insert spaces or other whitespace on either side of the operators ("=", "~", "!", "<", ">"). If your value should start with a space, include the value in quotation marks or protect the space with a backslash.

The sequence ":XY:" denotes the reference data field to search in. The names are mostly taken from the RIS specification. Possible field names are:

:TY:

Type of the reference.

:ID:

The unique identifier of a reference. This is the numeric identifier that refdb assigns to each new reference.

:CK:

The unique citation key of a reference. This is the alphanumeric string that was either supplied by the user or automatically generated by *refdb*.

:TI:, :T2:, :T3:, :TA:

The title of the reference, of the secondary title, and of the series title, respectively. :TA: performs a search in all title levels.

:AU:, :A2:, :A3:

The name of an author, of a secondary author/editor, and of a series author, respectively.

:PY:, :Y2:

The publication date and the secondary date, respectively.

:N1:

The notes that user can add to the reference.

:KW:

A keyword.

:RP:

The reprint status of the reference.

:AV:

The location of an offprint (physical, URL, or path)

:SP:

The start page.

:EP:

The end page.

:JO:, :JF:, :J1:, :J2:

The abbreviated name, the full name, the user abbreviation 1, and the user abbreviation 2 of a journal name, respectively.

:VL:

The volume number.

:ED:

The name of an editor.

:IS:

The issue (article) or chapter (book part) number.

:CY:

City of publication of a book.

:PB:

Name of the publishing company.

:U1: through :U5:

The user-defined fields 1 through 5. In certain cases, U5 contains the number of the paper copy of the reference (see the **addref** command).

:N2:

The abstract of the reference.

:SN:

The ISSN or ISBN number.

:M1: through :M3:

The Miscellaneous fields 1 through 3.

:L1: through :L4:

The link fields 1 through 4.

:AD:

The address of the contact person.

:UR:

The URL of a web page related to the reference.

In addition to the above field specifiers, there are a few that allow to retrieve references based on extended notes attached to them:

:NID:

The ID of an extended note.

:NCK:

The alphanumeric key of an extended note.

References are matched according to these rules:

- The alphanumerical fields are matched by (non-)equality to a literal string or to a regular expression. For literal matches the operators "=" and "!=" are accepted, denoting equality and non-equality, respectively. The *search-string* is a plain-text string.

Along the same lines, "~" and "!~" denote equality and non-equality for regular expression matches. The *search-string* can contain any legal characters and constructs as in standard Unix regular expressions. By default, the query matches if the search string is contained anywhere in the target string. If you need a left-match, a right-match, or a full match, use the regexp special characters "^" (match the beginning of a line) and "\$" (match the end of a line) to your needs. For further information about regular expressions, see the section regular expressions

Note: Some database engines, like SQLite, do not support Unix-style regular expressions. Use SQL regular expressions instead.

If you use regular expressions, be aware that you will have to escape characters with a special meaning if you want them to be matched literally. For further details, see the examples below.

- The numerical fields can be matched by equality, non-equality, or by greater-than/less-than comparisons. In these cases, use "!", ">", and "<" instead of the "=", respectively.
- If the *search-string* contains spaces, the whole string must be enclosed by single quotation marks or the spaces must be escaped with a backslash "\".
- Several search items can be combined by the Boolean operators "AND", "OR", " AND NOT". They can be grouped by brackets "()".

Note: If you use the operator "AND NOT", the search item it refers to has to be enclosed in brackets. See the examples below.

- The author (:AU:) and keyword (:KW:) searches allow an abbreviated syntax if you search for several authors or keywords at a time. The construct:

```
:AU:=name1 AND :AU:=name2 AND :AU:=name3 ...
```

can be written as:

```
:AU:='& name1 name2 name3 ...'
```

To specify an "OR" search, use "I" instead of "&".

Note: If an item in the search contains spaces, they have to be escaped by backslashes, as in "Amino\ Acid\ Sequence".

13.7.2. The notes query language

The `getnote` command uses essentially the same query language as described above. However, there is a different set of field specifiers that you can use to locate extended notes:

:NID:

The unique id of an extended note.

:NCK:

The unique citation key of an extended note.

:NPY:

The date of an extended note.

:NTI:

The title of an extended note.

:NKW:

A keyword. This is a keyword attached to a note in order to categorize the latter, similar to a keyword in a reference.

:KW:

A keyword. This is a keyword that the note is linked to, i.e. a keyword that the note was attached to in order to supply additional information.

:AU:

The name of an author or editor. Use this field specifier to locate notes that are linked to a particular author.

:JF:, :JO:, :J1:, :J2:

The full, abbreviated, or user-abbreviated name of a periodical. Use this field specifier to locate notes that are linked to a periodical.

:ID:

The id of a reference. Use this field specifier to locate notes linked to a particular reference.

:CK:

The citation key of a reference. Use this field specifier to locate notes linked to a particular reference.

13.7.3. Some example queries

This section shows a few example queries to help you get familiar with the syntax. If you are not familiar with the regular expressions used here, please peruse the regular expressions section. We will not use any of the fancy switches of the **getref** command here, so the output will always be a simple listing on the screen.

Note: These examples assume that your database engine supports Unix regular expressions. This holds true for MySQL and PostgreSQL, whereas SQLite uses the simpler SQL regular expressions instead.

We'll start with some easy queries. First we want to display a reference with a specific ID (25 in this example):

```
refdbc: getref :ID:=25
```

Next we want to list all references by a specific author. We'll use only the last name here. If several authors share this last name, we have to specify the initials as well, as shown in the second example. In the first example we use a regular expression match, denoted by the tilde operator. This obviates the need to know the full name precisely. The second example uses a literal match instead. Note the use of the caret "^" in the first example which makes sure that the name actually starts with the capital M. Otherwise, a last name like "DeMillerette" would match as well. This trick is not required in the second example as the literal match always implies a full match.

```
refdbc: getref :AU:~^Miller
refdbc: getref :AU:=Miller, J.D.
```

If Dr. Miller was a productive person, our previous query may have returned dozens of references. Now we try to filter out the paper or the papers that we really need. In the next example, we restrict the results to the years 1995 through 1999:

```
refdbc: getref :AU:~^Miller AND :PY:>1994 AND :PY:<2000
```

If this did not bring us close enough, we may try to include a coauthor:

```
refdbc: getref :AU:=~Miller AND :AU:~^Doe AND :PY:>1994 AND :PY:<2000
```

At this point we could narrow down the search by excluding other authors that often published with Dr. Miller, but are irrelevant here:

```
refdbc: getref :AU:~^Miller AND :AU:~^Doe AND NOT (:AU:~^Jones) AND :PY:>1994 AND :PY:<2000
```

Unfortunately, this is still a venerable list of publications. Now we try to include a few keywords. This is now a pretty complex query. It will return all references by the authors Miller and Doe between 1995 and 1999 with either the keyword "blood" or the keyword "animal" or the keywords "guanyl" and "cyclase", the latter only if both are present. The truncated spelling of "guanyl" ensures that both "guanylyl" and "guanylate" (which are interchangeable) will match. The funny expressions with the angle brackets ensure that the keywords will match regardless of whether they start with a capital letter or not.

```
refdbc: getref :AU:~^Miller AND :AU:~^Doe AND :PY:>1994 AND :PY:<2000 AND
(:KW:~[bB]lood OR :KW:~[aA]nimal OR (:KW:~[gG]uanyl AND :KW:~[cC]yclase))
```

And now for something completely different. If you've added a couple extended notes to your database, you can retrieve references that are attached to a specific extended note, e.g. to the note with the citation key "biochemistry1999":

```
refdbc: getref :NCK:=biochemistry1999
```

If you want to see all notes which are attached to a reference with the citation key "Miller1999", use the following command:

```
refdbc: getnote :CK:=Miller1999
```

Regular expressions may have unwanted side effects at times. Consider the keyword "52-67-5 (Penicillamine)" (a chemical name as used by the Pubmed (<http://ncbi.nlm.nih.gov>) database). Doing a literal match is straightforward:

```
refdbc: getref :KW:= '52-67-5 (Penicillamine)'
```

However, if we use the same argument for a regexp match, we won't get the desired results. The parentheses have a special meaning in regular expressions. Therefore we have to escape them if we want a literal match:

```
refdbc: getref :KW:~'\(Penicillamine\)'
```

Things are a little different again if you run a database engine that does not use Unix regular expressions, but SQL regular expressions instead. These know only '%' and '_' as special characters, and you have to escape them by doubling:

```
refdbc: getref :KW:~'100%%'
```

Tip: Remember that if you extend or modify a previous query, you don't have to retype everything: Just use the **up arrow key** to scroll through the previous commands, or use **Ctrl-r** to search for a specific query in the history.

13.8. Regular expressions

This section provides a brief overview over regular expressions. In the context of *refdb*, we have to deal with two flavors of regular expressions: Unix-style and SQL. The former are more important as we use them to write queries. The latter are used sparingly, e.g. to search the filenames of databases.

Note: Some database engines like SQLite do not support Unix-style regular expressions. You have to use SQL regular expressions in this case.

The difference between a literal match and a regular expression match is that the latter allows some “fuzziness” in the search string. The former requires that the search string and the search result match character by character. In simple words, regular expressions allow to search for strings which are similar to some extent, and you can exactly specify to which extent.

13.8.1. Unix-style regular expressions

Regular expressions distinguish between regular characters and special characters (meta characters). The simplest regular expressions actually don’t look like regular expressions, as the following example shows:

```
foo
```

This will search for the string “foo” at any position in the target elements. This would find strings like “foobar”, “lifoo”, or “lifooobar”. That is: if there are no meta characters, a simple string match is attempted, however at any position in the element. This is different from search strategies in some other databases where a full match or a left-match is attempted by default.

We can now replace one “o” in the above sample with a meta character. We use the “.” (dot) which matches any single character, including a newline, at that position:

```
f.o
```

This will find strings like “fao”, “fdo”, but as well all strings of the previous example.

Another very common meta character is the “*”, which matches zero or more instances of the *previous* character. Thus,

```
fo*
```

will now find things like “fo”, “foooo”, but also “fbar” and “lifooobar”. The meta character “+” is similar, but requires at least one instance of the *previous* character:

```
fo+
```

This would retrieve all strings of the last example except “fbar” as this contains the “o” zero times.

`fo?`

The questionmark meta character will retrieve either zero or one instances of the *previous* character. This would match “f” and “fo”, but not “foo”.

The meta characters “^” and “\$” are important to determine the relation of the search string to the line start or line end:

`^foo`

This will match “foo” only if it is located at the line start. Similarly,

`foo$`

will find “foo” only when it is located at the line end. If you combine these two like in the next example:

`^foo$`

“foo” will be found only if this is the complete element, starting and ending the line.

The following list briefly explains some more terms which are helpful in regular expressions.

()

Use the round brackets to group characters to a sequence. This is particularly useful with the above mentioned metacharacters *, +, and ?.

`(foo)*`

This will match zero or more instances of the sequence “foo”. It will find e.g. “foo” and “foofoo”, but not “fofo”.

[]

matches any single character between the brackets.

`[0-9]`

This will match any digit. Continuous ranges of characters can be indicated with a dash, as seen here.

[^]

matches any single character except the ones between the brackets

`[^abc]`

This will match any character except “a”, “b”, and “c”.

`\`

The backslash escapes the *following* meta character and treats it as a literal character.

`\.`

This will match only the dot instead of any single character.

`\{n,m\}`

This will find n to m repeats of the *previous* character.

`fo\{2,3\}`

This regular expression will find “foo” and “foo”, but not “fo” or “foooo”.

For further information about regular expressions, see the regex chapter in the MySQL documentation (<http://www.mysql.com>).

13.8.2. SQL regular expressions

SQL regular expressions are much simpler, as there are only two metacharacters:

`%`

matches any string

`_` (**underscore**)

matches any single character

In order to match a SQL regular expression special character literally, you have to escape it by doubling.

Chapter 14. redbib: the bibliography client

14.1. Overview

The purpose of this application is to create bibliographies based on an input file which contains a list of citations conforming to the CitationList XML DTD. This input file is thus not the document that you write but rather a list of citations that is extracted from your document. In the case of LaTeX/BibTeX bibliographies, the citations are listed in the `.aux` file which is automatically created by the LaTeX interpreter. In the other cases, the citation list is extracted from your document with (Open)Jade or xsltproc for SGML and XML documents, respectively.

Besides the bibliography data, redbib creates additional files in the case of SGML and XML documents. These are style sheet driver files which can be used for subsequent document transformations. They contain a few variable definitions to adapt the formatting to the required citation and bibliography style.

Note: While you certainly can run redbib directly, it is not recommended to do so in most cases. Some output, like the style specification driver files, needs post-processing to be fully usable. For your convenience, redbib ships with the runbib script which extracts the citation list, creates the bibliography file, and performs all necessary post-processing with a single command.

To make things even simpler, please have a look at the redbnd script. This script creates a skeleton SGML or XML file along with a customized `Makefile` which takes care of everything (except preparing coffee, that is).

14.2. Starting redbib

To run redbib, use the command:

```
redbib [-d database] [-D stylespec-directory] [-e log destination] [-E encoding] [-f  
stdin] [-h] [-i IP-address] [-l log level] [-L log file] [-m] [-n] [-N number] [-p port] [-q] [-S  
style] [-t output-type] [-T time] [-u name] [-v] [-V] [-w password] [-y confdir] [filename]
```

Remember that you don't have to specify all command-line options each time if you define the values in `.redbibrc`.

Use the `-d` option to specify the database that you want to work with. This will be the default database for all references unless the input file explicitly states the database for some or all citations.

redbib will create a style specification file for the subsequent transformation of your document with each run, unless you use the `-n` switch. These files will be stored in the directory that you specify with the `-D` option. Specify either a full path or `"."` to use the current working directory. The latter case is what you usually want if you run redbib from the directory where your LaTeX or SGML/XML document is stored. This is also the default if you do not specify a directory at all.

Use the `-E` option to select a useful output character encoding. If this option is not used, the bibliography data will use the character encoding of the database. See **man iconv_open** for a list of available encodings.

The `-f stdin` option is a crutch to make reading data from stdin possible on platforms that do not allow automatic detection of data on stdin, like Windows/Cygwin. On other platforms, `refdbib` automatically reads data from stdin if data are available.

If you pass the `-m` option to `refdbib`, missing references (i.e. cited references that are not in the database) will not cause an error. Processing scripts like `runbib` will thus continue regardless instead of throwing in the towel.

The `-n` option instructs `refdbib` not to create style specification files. Use this option if you are sure that a current style specification file already exists. This saves about as much computing time as you need to key in this option.

Use the `-N` option to specify where the numbering of the references is supposed to start. The default is 1. This option comes in handy if you need to cobble together composite bibliographies or per-chapter bibliographies that still need to be numbered consecutively.

The `-S` option specifies the bibliography style. This controls the formatting of the bibliography and the in-text citations when the source document is processed. Possible values can be obtained with the **liststyle** command in `refdbc`.

The `-t` option selects the type of output. Use "db31" to generate DocBook SGML bibliographies, "db31x" for DocBook XML bibliographies, "teix" for TEI XML bibliographies, and "bibtex" for BibTeX bibliographies. The type of output also determines the type of style specification file, if any, that will be generated in addition to the bibliography for formatting purposes. This is only a matter of concern if you want to process a DocBook XML document with the DSSSL stylesheets: In this case you should use "db31" with this option. The SGML bibliography element is also a valid XML element, but you will get a DSSSL driver file instead of a XSL driver file when you use "db31x".

Note: In the current implementation, the `-t teix` option will also return a DocBook bibliography which needs to be transformed to a TEI bibliography with the `bibdb2tei.xsl` stylesheet.

The purpose of all other command-line switches is explained in the section common command-line options.

`refdbib` accepts the data either at stdin or from a data file which is specified on the command line. All output will be written to stdout.

The exit code is 0 if everything went fine and >0 if an error occurred.

14.3. Examples

The first example shows how to create a DocBook SGML bibliography file.

```
$~ refdbib -d myrefs -S "Br.J.Pharmacol." -t db31 -D "." mypaper.id.xml > mypaper.bib.sgml
```

This command will use the database “myrefs” to retrieve the references defined in `mypaper.id.xml`. They will be formatted according to the bibliography style called “Br.J.Pharmacol.” and will be redirected into the bibliography file `mypaper.bib.sgml`. The DSSSL driver file (it will be automatically named after the bibliography style, that is `Br.J.Pharmacol.dsl`) will be stored in the current working directory. For further information how to generate the citation listing in `mypaper.id.xml`, see DocBook.

The second example shows how to create the BibTeX bibliography from your LaTeX document (it is assumed that you ran **latex** at least once before this command. See LaTeX/BibTeX for further explanations).

```
$~ refdbib -d myrefs -S "name" -t bibtex mypaper.aux > mypaper.bib
```

This command will use the database “myrefs” to retrieve the references defined in `mypaper.aux`. The intermediate bibliography database will be stored in `mypaper.bib` and will serve as an input file for **bibtex**.

Note: For the sake of consistency with **bibtex**, it is possible to specify the auxiliary file without the `.aux` extension (`mypaper` in the above example).

If you are working on a long document that cites the same references over and over again, it may be prudent to preprocess the `.aux` file in order to eliminate duplicates (duplicates do not confuse **bibtex** but they waste space):

```
$~ sort mypaper.aux | uniq | refdbib -d myrefs -S "name" -t bibtex > mypaper.bib
```

Note: The `runbib` script does exactly this kind of preprocessing automatically.

14.4. The refdbibrc variables

Table 14-1. refdbibrc

Variable	Default	Comment
refdblib	(none)	The path of a directory containing shareable files like DTDs, HTML templates etc.

Variable	Default	Comment
defaultdb	(none)	The default database. redbib will use this database unless you specify the databases in the citation elements of your documents.
pager	stdout	The command line of a pager that accepts the output of redb on stdin to allow scrolling and other nifty things. “stdout” sends the data to stdout.
passwd	*	The password which is used for authentication with the database server. It is potentially evil to store unencrypted passwords in disk files. At least make sure that the init file is not readable for anyone else. The default setting causes redbib to ask for your password interactively.
port	9734	The port on which redbd listens. Change this for all clients and the server if this value interferes with another program using this port.
serverip	127.0.0.1	The IP address or hostname of the machine where redbd runs. Use the default (localhost) address if the clients and redbd run on the same machine.
timeout	180	The timeout in seconds. After this time has elapsed, a stalled connection is taken down. Increase this value if you encounter frequent timeout errors due to high network traffic or redbd's overload.
username	login name	The username which is used for authentication with the database server. This may be different from the login name of the user.
verbose	f	Set this to t if you prefer verbose error messages.

Variable	Default	Comment
logfile	/var/log/refdbib.log	The full path of a custom log file. This is used only if logdest is set appropriately.
logdest	1	The destination of the log information. 0 = print to stderr; 1 = use the syslog facility; 2 = use a custom logfile. The latter needs a proper setting of logfile.
loglevel	6	The log level up to which messages will be sent. A low setting (0) allows only the most important messages, a high setting (7) allows all messages including debug messages. -1 means nothing will be logged.
outtype	db3l	The type of output generated. Use <i>db3l</i> for DocBook SGML bibliographies, <i>db3lx</i> for DocBook XML bibliographies, <i>teix</i> for TEI XML bibliographies, and <i>bibtex</i> for BibTeX bibliographies.
outformat	(none)	The bibliographic style to be used for the output. This is the name of a style as it was previously added to the database.
stylespecdir	.	A path to a directory (including the trailing directory separator) that will receive the stylesheet driver files. The default setting will direct the driver files to the current working directory that most likely contains the input files. It should rarely be necessary to use a different setting.

Variable	Default	Comment
startnumber	1	The number where the reference numbering starts at. This option is mostly useful for compiling advanced bibliographies or for C boneheads who insist that counting starts at zero.
toencoding	(the database encoding)	The character encoding for the bibliography output. If this is not specified, the data will use the same encoding as the database.
ignore_missing	f	If this is set to "f", missing references (i.e. cited but not in the database) will throw an error. If set to "t", you'll get a warning but missing references will not cause refdbib to return an error.

Chapter 15. Reference data conversion tools

Whenever you have an electronic source for reference data, you should use these instead of typing datasets from scratch. Reference data come in a variety of formats. Therefore refdb ships with a few conversion utilities which are discussed in this chapter.

The following utilities create tagged RIS data from the input data. The final section of this chapter gives some hints about converting data to the risx XML format that refdb supports since version 0.9.3.

15.1. dos2unix

This simple shell script converts DOS/Windows files with CR/LF line endings to Unix files with LF line endings. It is nothing more than a wrapper for **tr**. Use it to convert files that you downloaded or created in a plain DOS/Windows environment (if you created them with Cygwin tools or a configurable editor like NTEmacs, they should not need this conversion). To convert files, use the command:

```
~/# dos2unix file1 [file2...]
```

All files which are passed as arguments will be converted and the results will be written back to the same files.

If this is not successful on Windows, you may have to fiddle with the text-vs.-binary mode setting in Cygwin. Try adding *binmode* to your CYGWIN environment variable.

15.2. med2ris

This Perl script converts Pubmed reference data into RIS data. The converter understands both the tagged Pubmed format (which superficially resembles RIS) and the XML format according to the PubMedArticle DTD (http://www.ncbi.nlm.nih.gov/entrez/query/DTD/pubmed_020114.dtd). In most cases med2ris is able to automatically detect the input data type.

15.2.1. Starting med2ris

Start the script with the following command:

```
[perl] med2ris [-e dest] [-f enc] [-h] [-i] [-l level] [-L logfile] [-o file] [-O file] [-q] [-t enc] [-T type] [-y path] [infile...]
```

Note: Specifying the command interpreter **perl** on the command line is not necessary if it is in the default location `/usr/bin/perl`.

The `-e` option takes either a numeric (0|1|2) or a symbolic (stderr|syslog|file) argument to specify the log destination.

The `-f` and `-t` options select the input and output character encoding, respectively. Supported encodings are platform-dependent and can usually be retrieved by running `man iconv` or `man iconv_open`. If no encodings are specified, "ISO-8859-1" aka Latin-1 is assumed for both input and output.

The `-h` displays a brief usage message.

Set the `-i` option to output additional information about unknown or unused tags.

Use the `-l` option to set the log level to a numeric value between 0 and 7 or to a symbolic value (alert|crit|err|warning|notice|info|debug). If the log destination is "file", the `-L` option specifies the full path of a custom log file.

The `-o` and `-O` options cause **med2ris** to write the output data into a file. The lowercase `-o` option will overwrite any existing file of the same name while the uppercase `-O` option will append the output to an existing file. If none of these options is used, the output will be written to stdout.

The `-q` option will cause **med2ris** to skip the configuration file which is mainly useful for debugging purposes.

Use the `-T` option to override the automatic input data type detection. Possible values for type are "xml" and "tag" for the XML and tagged data formats, respectively.

The `-y` switch can be used to specify the location of the `refdb` shared data in case the automatic script configuration is not appropriate on your system.

The input data are read from stdin unless one or more filenames are specified on the command line. In the latter case, the output generated from all files will be sent to stdout or to the output file.

The following examples show the usage of **med2ris** for file-based and stream-based in/output, respectively.

```
~# perl med2ris -o out.ris pm*
```

This will convert all files in the current directory starting with `pm` and write the output into `out.ris`, overwriting any existing file with the same name.

Note: You can leave out the "perl" in the above command if your Perl interpreter is in the default location `/usr/bin/perl`, as shown in the next example.

```
~# med2ris -f "ISO-8859-1" -t "UTF-8" < pm001.txt >> out.ris
```

This will send the contents of `pm001.txt` to **med2ris** and convert the contents. The result will be appended to the file `out.ris`. The input data are assumed to be Latin-1, whereas the output will be Unicode.

15.2.2. The med2ris configuration variables

med2ris evaluates the file `med2risrc` to set its default values.

Table 15-1. med2risrc

Variable	Default	Comment
outfile	(none)	The default output file name.
outappend	t	Determines whether output is appended (t) to an existing file or overwrites (f) an existing file.
unmapped	t	If set to t, unknown tags in the input data will be output following a <unmapped> tag; the resulting data can be inspected and then be sent through sed to strip off these additional lines. If set to f, unknown tags will be gracefully ignored.
from_enc	ISO-8859-1	The character encoding of the input data
to_enc	ISO-8859-1	The character encoding of the output data
logfile	/var/log/med2ris.log	The full path of a custom log file. This is used only if logdest is set appropriately.
logdest	1	The destination of the log information. 0 = print to stderr; 1 = use the syslog facility; 2 = use a custom logfile. The latter needs a proper setting of logfile.
loglevel	6	The log level up to which messages will be sent. A low setting (0) allows only the most important messages, a high setting (7) allows all messages including debug messages. -1 means nothing will be logged.

15.2.3. med2ris' behind-the-scenes data mangling

Keywords with multiple MeSH subheadings are split into multiple keywords with one MeSH subheading each. This simplifies searching for MeSH subheadings greatly.

med2ris does not validate the input files. That is, the input files must stick to the rules of the data sources, otherwise the conversion results are not predictable.

15.3. nmed2ris

Note: nmed2ris is doomed to disappear from the refdb distribution as most of the functionality (and then some) is available through the med2ris script these days. Don't rely on the existence of this program in future releases. Don't use it. Don't read on. Skip this section.

This input filter accepts a variety of Medline tagged file formats and converts them to RIS format. To import the Pubmed XML format, please see med2ris in the next section. The input is accepted in DOS (with CR/LF line endings) and Unix (LF) formats.

15.3.1. Starting nmed2ris

Start nmed2ris with the command:

```
nmed2ris [-e log-destination] [-h] [-i t|f] [-l log-level] [-L log-file] [-o output-file | -O output-file] [-q] [-s data-source] [-v] [-y confdir] [-f input-file | file]
```

Remember that you don't have to specify all these options each time if you define the values in nmed2risrc.

The `-e` option defines the destination of log output. In order for log output to appear at all, the log level has to be specified correctly with the `-l` option. A *log-destination* argument of 0 directs log output to stderr, 1 uses the syslog facility, 2 uses a custom log file. For the latter to work you have to specify a log filename with the `-L` option.

With the `-h` option nmed2ris displays a brief help screen and exits.

Use the `-i` option to determine what to do with unknown tags in the source files. An unknown tag may lead to unwanted loss of information. Use the value *t* to simply ignore unknown tags. If you use *f* instead, any unknown tag will generate an error.

The `-l` option determines the maximum log level that a log message may have to be logged. If you specify a high level (≤ 7), all sorts of messages including debug messages are logged. If you specify a low level (≥ 0), only critical errors are logged. Specify `-1` to disable logging.

The `-L` option specifies a filename which is used as a custom log file if the `-e` option is set appropriately.

The `-o` and `-O` options specify a filename where the output gets written to or appended, respectively. If neither of these options is used, the output is written to stdout.

Use the `-q` option to temporarily switch off the settings in the init files. **nmed2ris** will then use the compile-time defaults unless you specify things with the command line switches (useful for debugging configuration file settings).

Specify the data source with the `-s` option. Valid arguments are BM (BioMedNet), CC (Current Contents on disk), GM (Grateful Med), HG (HealthGate), KF (Knowledge Finder), PM (PubMed).

`-v` prints the version information and brief licensing information, then exits.

Use the `-y` to specify the directory where the global configuration files are.

Note: By default, all refdb applications look for their configuration files in a directory that is specified during the configure step when building the package. That is, you don't need the `-y` option unless you use precompiled binaries in unusual locations, e.g. by relocating a rpm package.

All other command line parameters will be interpreted as input filenames. **nmed2ris** can read the incoming data either from these files or from stdin. If data are available at stdin, the filename arguments will be ignored. The output is sent to stdout unless you use one of the `-o` and `-O` options, so you can either view the result by piping into a pager or redirect the data into a file. Of course it is also possible to directly pipe the result into refdbc but it may be prudent to manually check the output before sending something to refdbc that you may later regret.

All other command line parameters will be interpreted as input filenames.

The following examples show the usage of **nmed2ris** for file-based and stream-based in/output, respectively.

```
~# nmed2ris -o out.ris pm*
```

This will convert all files in the current directory starting with `pm` and write the output into `out.ris`, overwriting any existing file with the same name.

```
~# nmed2ris -s PM < pm001.txt >> out.ris
```

This will direct the contents of `pm001.txt` to stdin of **nmed2ris** and convert the contents. The result will be appended to the file `out.ris`.

15.3.2. The nmed2ris configuration variables

nmed2ris reads the file `nmed2risrc` to set its defaults.

Table 15-2. **nmed2risrc**

Variable	Default	Comment
----------	---------	---------

Variable	Default	Comment
medsource	PUBMED	The default source of input data.
outfile	(none)	The default output file name.
outappend	t	Determines whether output is appended (t) to an existing file or overwrites (f) an existing file.
ignoretag	t	If set to t, unknown tags in the input data will be silently ignored. If set to f, each unknown tag will generate an error message.
logfile	/var/log/nmed2ris.log	The full path of a custom log file. This is used only if logdest is set appropriately.
logdest	1	The destination of the log information. 0 = print to stderr; 1 = use the syslog facility; 2 = use a custom logfile. The latter needs a proper setting of logfile.
loglevel	6	The log level up to which messages will be sent. A low setting (0) allows only the most important messages, a high setting (7) allows all messages including debug messages. -1 means nothing will be logged.

15.3.3. nmed2ris' behind-the-scenes data mangling

While the primary purpose of **nmed2ris** is the conversion of various Medline tagged formats to the RIS format digestable for refdb, it does some useful things on the fly:

- Regardless of the number of original input files, you'll have to deal with only one output file or output stream at stdout.
- Different Medline providers use different formats for the MeSH subheadings. All supported formats are consolidated into a single format to get a consistent database.
- Keywords with multiple MeSH subheadings are split into multiple keywords with one MeSH subheading each. This simplifies searching for MeSH subheadings greatly.

nmed2ris is no parser and does not validate the input files, though. That is, the input files must stick to the rules of the data sources, otherwise the conversion results are not predictable. **nmed2ris** will act according to "garbage in, garbage out" in most cases.

15.3.4. Data sources

nmed2ris currently supports data from the following source:

- PubMed (<http://www.ncbi.nlm.nih.gov/PubMed/>)

nmed2ris also contains code to import data from other sources. This code may be out of date as it is currently not maintained.

- BioMedNet (<http://biomednet.com>)
- Current Contents (<http://www.isinet.com>) (no MEDLINE data)
- Grateful Med (<http://igm.nlm.nih.gov/>)
- HealthGate (<http://www.healthgate.com>)
- Knowledge Finder (<http://www.kfinder.com>)

The data source can be explicitly specified with an *medsource* entry in the init file or with the `-s` command line option. Either of these should be specified if you read the data from stdin, otherwise **nmed2ris** defaults to PubMed. Alternatively, you can use a semi-automatic datasource recognition by filename prefixes (this clearly doesn't work for input on stdin). When downloading the files from any of the online sources, simply prefix the filenames with a case-insensitive two-letter code denoting the datasource:

BM

BioMedNet

CC

Current Contents

GM

Grateful Med

HG

HealthGate

KF

KnowledgeFinder

PM

PubMed

Thus `pm001.txt` would be recognized as a *PubMed* input file. A *PubMed* file not starting with `pm` would need either the command line switch `-s PM` or the init file setting `medsource PM`.

15.4. bib2ris

bib2ris converts BibTeX bibliography files into RIS files. The filter is implemented using Greg Ward's outstanding `btparse` library.

Unfortunately the concepts underlying BibTeX and RIS bibliographic data are quite different so that BibTeX data do not readily lend themselves to a clean conversion to the RIS format. This is not meant as an excuse to provide a bad filter but you should be aware that a few compile-time assumptions have to be made in order to get reasonable results. In any case, as the data models differ considerably, a loss-free interconversion between the two data types is not possible: If you convert a BibTeX bibliography file to RIS and then back, the result will differ considerably from your input.

There are basically two ways how to handle BibTeX data with `refdb`:

1. Convert all entries to plain text. This will allow you to work with your data just as with "native" RIS data, i.e. all field values in the output of the `refdb` backends will be plain text as well.
2. Keep the TeX formatting in the entries. This will allow you to make use of TeX commands and formatting stuff in the BibTeX bibliography output, but it'll be a bit strange to work with these data in the rest of `refdb`. When formulating queries you will have to take account of the TeX magic, and this stuff will also show up in all other output (screen, HTML, DocBook etc).

There may be better support for this situation in future releases of `refdb`. Currently the rule of thumb is: If you're interested only in BibTeX bibliographies, keep the formatting. If you're interested in generating both BibTeX and DocBook bibliographies or if you're mainly interested to maintain an easily accessible reference database, strip off the TeX formatting. This is best done with the supplied **tex2mail** script which will be discussed shortly.

That said, you may still be interested to see how it works.

15.4.1. Starting bib2ris

Start **bib2ris** with the command:

```
bib2ris [-e log-destination] [-h] [-j] [-l log-level] [-L log-file] [-q] [-s separator] [-v] [-y confdir] [file]
```

Remember that you don't have to specify all these options each time if you define the values in `bib2risrc`.

The `-e` option defines the destination of log output. In order for log output to appear at all, the log level has to be specified correctly with the `-l` option. A *log-destination* argument of 0 directs log output to `stderr`, 1 uses the `syslog` facility, 2 uses a custom log file. For the latter to work you have to specify a log filename with the `-L` option.

With the `-h` option `bib2ris` displays a brief help screen and exits.

Use the `-j` option to force `bib2ris` to use "JO" RIS fields in all cases. If this option is not used, **bib2ris** tries to infer whether a journal name is an abbreviation or not. If the string contains at least one period, "JO" will be used, otherwise "JF" will be used.

The `-l` option determines the maximum log level that a log message may have to be logged. If you specify a high level (≤ 7), all sorts of messages including debug messages are logged. If you specify a low level (≥ 0), only critical errors are logged. Specify `-1` to disable logging.

The `-L` option specifies a filename which is used as a custom log file if the `-e` option is set appropriately.

Note: The underlying `btparse` library sends some warnings and errors directly to `stderr`. Currently (i.e. without patching `btparse`) this behaviour cannot be controlled with the `-e`, `-l`, and `-L` switches. If you want to log these messages to a file as well you will have to employ some shell magic to redirect the output.

Use the `-q` option to temporarily switch off the settings in the init files. `bib2ris` will then use the compile-time defaults unless you specify things with the command line switches (useful for debugging configuration file settings).

The `-s` specifies the delimiter which separates individual keywords in a non-standard keyword field. Use the string `"spc"` for whitespace-separated lists (spaces and tabs).

`-v` prints the version information and brief licensing information, then exits.

Use the `-y` to specify the directory where the global configuration files are.

Note: By default, all `refdb` applications look for their configuration files in a directory that is specified during the configure step when building the package. That is, you don't need the `-y` option unless you use precompiled binaries in unusual locations, e.g. by relocating a rpm package.

All other command line parameters will be interpreted as input filenames. `bib2ris` can read the incoming data either from these files or from `stdin`. If data are available at `stdin`, the filename arguments will be ignored. The output is always sent to `stdout`, so you can either view the result by piping into a pager or redirect the data into a file. Of course it is also possible to directly pipe the result into `refdb` but it may be prudent to manually check the output before sending something to `refdb` that you may later regret.

The exit code of **bib2ris** indicates what went wrong in general (the details can be found in the log output). The code is the sum of the following error values:

Table 15-3. bib2ris exit codes

code	explanation
1	general error; includes out of memory situations and invalid command-line options
2	incomplete entry (at least one essential field in an entry was missing)
4	unknown field name
8	unknown publication type
16	invalid BibTeX->RIS type mapping

code	explanation
32	parse error; includes file access errors

As an example, if **bib2ris** exits with the error code 18, simple math would tell you that there was at least one error in the BibTeX-to-RIS mapping (16), most likely an invalid RIS tag, and at least one incomplete entry (2), the sum of which yields 18.

Note: Under some (really bad) error conditions the underlying btparse library exits without returning control to **bib2ris**. In that case the exit code is determined by btparse, not by **bib2ris**.

The following examples show how **bib2ris** reads data from stdin or from input files, respectively.

```
~# bib2ris *.bib | less
```

This command will convert all `.bib` files in the current directory and display the result in a pager.

```
~# bib2ris < foo.bib > foo.ris
```

This command reads the data via redirection from `foo.bib` and redirects the output into the file `foo.ris`.

15.4.2. The bib2ris variables

bib2ris evaluates the file `bib2risrc` to initialize itself.

Table 15-4. bib2risrc

Variable	Default	Comment
logfile	/var/log/bib2ris.log	The full path of a custom log file. This is used only if logdest is set appropriately.
logdest	1	The destination of the log information. 0 = print to stderr; 1 = use the syslog facility; 2 = use a custom logfile. The latter needs a proper setting of logfile.
loglevel	6	The log level up to which messages will be sent. A low setting (0) allows only the most important messages, a high setting (7) allows all messages including debug messages. -1 means nothing will be logged.

Variable	Default	Comment
abbrevfirst	t	If this option is set to "t", the first names of all authors and editors will be abbreviated to the initials. If set to "f", the first names will be used as they are found in the BibTeX bibliography file.
listsep	;	This is the delimiter which separates individual keywords in a non-standard keyword field. Use the string "spc" for whitespace-separated lists (spaces and tabs).
forcejabbrev	f	If this is set to "t", journal names will be wrapped in RIS "JO" entries. If it is set to "f", bib2ris will use "JO" entries only if the journal name contains at least one period, otherwise it will use "JF".
maparticle	JOUR	map the BibTeX article publication type to a RIS type
mapbook	BOOK	map the BibTeX book publication type to a RIS type
mapbooklet	PAMP	map the BibTeX booklet publication type to a RIS type
mapconference	CHAP	map the BibTeX conference publication type to a RIS type
mapinbook	CHAP	map the BibTeX inbook publication type to a RIS type
mapincollection	CHAP	map the BibTeX incollection publication type to a RIS type
mapinproceedings	CHAP	map the BibTeX inproceedings publication type to a RIS type
mapmanual	BOOK	map the BibTeX manual publication type to a RIS type
mapmastersthesis	THES	map the BibTeX mastersthesis publication type to a RIS type
mapmisc	GEN	map the BibTeX misc publication type to a RIS type
mapphdthesis	THES	map the BibTeX phdthesis publication type to a RIS type

Variable	Default	Comment
mapproceedings	CONF	map the BibTeX proceedings publication type to a RIS type
maptechreport	RPRT	map the BibTeX techreport publication type to a RIS type
mapunpublished	UNPB	map the BibTeX unpublished publication type to a RIS type
nsf_xyz	(none)	You can specify an unlimited number of these entries to map non-standard BibTeX fields to RIS tags. The BibTeX field name in this variable has to be in lowercase, regardless of the case in your input data (bib2ris treats field names as case-insensitive). The two-letter RIS tag has to be in uppercase. E.g. to map your BibTeX "Abstract" field to the RIS "N2" tag, the entry would read: "nsf_abstract N2".

Table 15-5. bib2risrc

Variable	Default	Comment
refdblib	(none)	The path of the directory containing shareable refdb files like DTDs, HTML templates etc.
logfile	/var/log/bib2ris.log	The full path of a custom log file. This is used only if logdest is set appropriately.
logdest	1	The destination of the log information. 0 = print to stderr; 1 = use the syslog facility; 2 = use a custom logfile. The latter needs a proper setting of logfile.
loglevel	6	The log level up to which messages will be sent. A low setting (0) allows only the most important messages, a high setting (7) allows all messages including debug messages. -1 means nothing will be logged.

Variable	Default	Comment
abbrevfirst	t	If this option is set to "t", the first names of all authors and editors will be abbreviated to the initials. If set to "f", the first names will be used as they are found in the BibTeX bibliography file.
listsep	;	This is the delimiter which separates individual keywords in a non-standard keyword field. Use the string "spc" for whitespace-separated lists (spaces and tabs).
forcejabbrev	f	If this is set to "t", journal names will be wrapped in RIS "JO" entries. If it is set to "f", bib2ris will use "JO" entries only if the journal name contains at least one period, otherwise it will use "JF".
maparticle	JOUR	map the BibTeX article publication type to a RIS type
mapbook	BOOK	map the BibTeX book publication type to a RIS type
mapbooklet	PAMP	map the BibTeX booklet publication type to a RIS type
mapconference	CHAP	map the BibTeX conference publication type to a RIS type
mapinbook	CHAP	map the BibTeX inbook publication type to a RIS type
mapincollection	CHAP	map the BibTeX incollection publication type to a RIS type
mapinproceedings	CHAP	map the BibTeX inproceedings publication type to a RIS type
mapmanual	BOOK	map the BibTeX manual publication type to a RIS type
mapmastersthesis	THES	map the BibTeX mastersthesis publication type to a RIS type
mapmisc	GEN	map the BibTeX misc publication type to a RIS type
mapphdthesis	THES	map the BibTeX phdthesis publication type to a RIS type

Variable	Default	Comment
mapproceedings	CONF	map the BibTeX proceedings publication type to a RIS type
maptechreport	RPRT	map the BibTeX techreport publication type to a RIS type
mapunpublished	UNPB	map the BibTeX unpublished publication type to a RIS type
nsf_xyz	(none)	You can specify an unlimited number of these entries to map non-standard BibTeX fields to RIS tags. The BibTeX field name in this variable has to be in lowercase, regardless of the case in your input data (bib2ris treats field names as case-insensitive). The two-letter RIS tag has to be in uppercase. E.g. to map your BibTeX "Abstract" field to the RIS "N2" tag, the entry would read: "nsf_abstract N2".

15.4.3. bib2ris' data mangling

This section provides a few hints about the data conversion itself and the BibTeX format requirements.

- The parsing of the input data is done by the `btparse` library. All limitations of that library apply to `bib2ris` as well. This applies very specifically to two hardcoded settings in `btparse` which, simply put, limit the size and complexity (in terms of macros) of an input file that `btparse` can handle. If you run into this kind of problem (I had to pull a 2 MB BibTeX bibliography from the net in order to verify this limit) you should increase the values of `NUM_MACROS` and `STRING_SIZE` in the source file `macros.c` and recompile the `btparse` library.
- All entry names and field names in the BibTeX input file are treated as case-insensitive, i.e. "BoOk" is the same as "book" and "AUTHOR" is the same as "aUthoR".
- The entries are checked for completeness. An error is generated if an entry lacks fields which are considered essential for the particular publication type.
- Non-standard fields can be imported in addition to the predefined BibTeX fields. Create an entry for each non-standard BibTeX field name that your input data use in your `bib2ris` configuration file. The data are handled differently based on the type of RIS field they are imported to. If the data are imported to the RIS fields AD, N1, or N2, which basically have an unlimited size, all occurrences of these fields will be concatenated into a single AD, N1, or N2 tag line, respectively. If the data are mapped to the RIS KW field, the string will be tokenized based on the list separator specified in the

listsep configuration variable. Each token will be written as a separate KW tag line. A special case is the RIS pseudo-field "PY.day". Data imported to this tag are integrated as the day part in the publication date tag line "PY" (year and month, but not day, are standard BibTeX fields and are recognized by default). All other fields will be printed with their requested RIS tag. It is at the discretion of any RIS importing application to decide what to do with duplicate tag lines. Multiples are allowed for author tags (AU, A2, A3) and the keyword tag (KW). refdb will use the *last* occurrence of a tag line that does not allow multiple occurrences.

- Abbreviated journal names are detected only if they use periods. E.g. "J. Biol. Chem." will be mapped to a "JO" RIS element whereas "J Biol Chem" will be (incorrectly) mapped to a "JF" element ("Journal of Biological Chemistry" would correctly end up here too). Spaces after periods are optional. To capture "J Biol Chem" in a "JO" element, use the `-j` command line option or the "forceabbrev" configuration file variable.
- The mapping of BibTeX publication types (book, inproceedings...) to RIS types as specified in the configuration file is checked for valid RIS types. If an invalid RIS type is specified, an error is generated and the compile-time default is used instead.
- By default the first names of authors and editors are not abbreviated. If you wish you can configure **bib2ris** to abbreviate first and middle names.

15.4.4. Post-processing with tex2mail

refdb ships with a slightly modified version of the **tex2mail** Perl script. The original purpose of this script is to convert (La)TeX input into a human-readable plain text file, taking care of various mathematical commands which can be rendered in multi-line output. In lieu of a better way to provide something useful in no time I hacked this script to generate suitable RIS output when used with the proper command line switches. Without the `-ris` switch the script behaves just like the original tex2mail script. The purpose of this script in the context of refdb is to strip TeX commands and constructs from the RIS output that bib2ris generates.

Warning

- This script is really a quick hack. It will be replaced by something more dedicated to its purpose (at least I'll maintain this illusion for the time being).
- If you have LaTeX math formulas somewhere in the field values, strange and wondrous things are likely to happen. You will have to manually fix the output.

Run the script with the following command, assuming that `foo.ris` is the output that you generated from your BibTeX bibliography file with the help of bib2ris:

```
~# tex2mail -noindent -ragged -linelength 65535 -ris < foo.ris > foo-notex.ris
```

The argument of the `-linelength` option should be large enough to display each field in a single line, otherwise `tex2mail` tries to generate some simple layout which will screw up the RIS file.

15.5. en2ris

EndNote is a commercial bibliography package available for Windows and MacOS/OSX. It claims to export RIS data but this doesn't quite work out. In order to transfer existing reference data from EndNote to `refdb`, you'll have to process your data through this script.

15.5.1. Starting en2ris

Start the script with the following command:

```
[perl] en2ris [-e dest] [-f enc] [-h] [-l level] [-L logfile] [-o file] [-O file] [-q] [-t enc] [-y path]
```

Note: Specifying the command interpreter **perl** on the command line is not necessary if it is in the default location `/usr/bin/perl`.

The `-e` option takes either a numeric (0|1|2) or a symbolic (`stderr|syslog|file`) argument to specify the log destination.

The `-f` and `-t` options select the input and output character encoding, respectively. Supported encodings are platform-dependent and can usually be retrieved by running **man iconv** or **man iconv_open**. If no encodings are specified, "ISO-8859-1" aka Latin-1 is assumed for both input and output.

The `-h` displays a brief usage message.

Use the `-l` option to set the log level to a numeric value between 0 and 7 or to a symbolic value (`alert|crit|err|warning|notice|info|debug`). If the log destination is "file", the `-L` option specifies the full path of a custom log file.

The `-o` and `-O` options cause **med2ris** to write the output data into a file. The lowercase `-o` option will overwrite any existing file of the same name while the uppercase `-O` option will append the output to an existing file. If none of these options is used, the output will be written to `stdout`.

The `-q` option will cause `med2ris` to skip the configuration file which is mainly useful for debugging purposes.

The `-y` switch can be used to specify the location of the `refdb` shared data in case the automatic script configuration is not appropriate on your system.

The input data are read from `stdin`. The output is sent to `stdout` unless you use one of the `-o` and `-O` options.

The following example shows the usage of **en2ris**:


```
~# perl en2ris -f "ISO-8859-1" -t "UTF-8" -o out.ris < endnote.ris
```

This will convert the EndNote RIS data in `endnote.ris` and write the output into `out.ris`, overwriting any existing file with the same name. The input data are assumed to be Latin-1, whereas the output will be Unicode.

Note: You can leave out the "perl" in the above command if your Perl interpreter is in the default location `/usr/bin/perl`

15.5.2. The en2ris configuration variables

Table 15-6. en2risrc

Variable	Default	Comment
outfile	(none)	The default output file name.
outappend	t	Determines whether output is appended (t) to an existing file or overwrites (f) an existing file.
from_enc	ISO-8859-1	The character encoding of the input data
to_enc	ISO-8859-1	The character encoding of the output data
logfile	/var/log/med2ris.log	The full path of a custom log file. This is used only if logdest is set appropriately.
logdest	1	The destination of the log information. 0 = print to stderr; 1 = use the syslog facility; 2 = use a custom logfile. The latter needs a proper setting of logfile.
loglevel	6	The log level up to which messages will be sent. A low setting (0) allows only the most important messages, a high setting (7) allows all messages including debug messages. -1 means nothing will be logged.

15.5.3. en2ris' behind-the-scenes data mangling

en2ris fixes a couple of problems found in RIS data exported from EndNote. The main issues are the incomplete date formats, the export of page ranges into a single "SP" tag line, and the export of keywords as a list into a single "KW" tag line.

en2ris does not validate the input files. That is, the input files must stick to the rules of the data sources, otherwise the conversion results are not predictable.

15.6. db2ris

This input filter attempts to convert bibliographic data in the DocBook format into the RIS format. A similar complaint as with the BibTeX import filter applies here: The design and purpose of the DocBook bibliography information is too different from the RIS format to warrant a simple and straightforward conversion, let alone a bi-directional one. The DocBook bibliography definition allows for a lot of freedom how to encode your information. To accommodate as wide a range of uses and abuses of the bibliographic elements as possible, this import filter is implemented as a DSSSL stylesheet rather than as a C application. This allows you to quickly change or extend the stylesheet to adapt it to your needs. The costs of maintaining several local copies with different modifications is also much lower than with a compiled program.

15.6.1. Starting db2ris

We use a shell script to run the `db2ris.dsl` stylesheet in our favourite DSSSL engine and post-process the output. The reason for this post-processing is the fact that Jade and OpenJade lack an official plain-text backend (Jade allows to abuse the SGML backend to create plain-text output but this does no longer work with OpenJade). We have to use the SGML backend and create elements, so each RIS output line is wrapped in SGML-like tags which have to be stripped off afterwards. Besides that, the use of a shell script allows us to keep some settings in configuration files.

Note: The `-a` and `-r` option and the corresponding init file variables `authorlong` and `defaultreftype` require OpenJade. Jade does not support passing arbitrary variable values on the command line. If you use Jade, please modify the `db2ris.dsl` stylesheet directly or create appropriate driver files.

```
db2ris [-a] [-e log-destination] [-h] [-L log-file] [-o file] [-O file] [-r reftype] [file...]
```

Remember that you don't have to specify all these options each time if you define the values in `db2risrc`. Use the `-a` switch to use full first- and othernames if they are present in the DocBook source. By default, all first- and othernames will be abbreviated.

You can send error messages to a custom log file by setting the `-e` option to "2". By default, error messages appear on `stderr`.

Set the path to a custom log file with the `-L` option. This file will only be used if the `-e` option is set accordingly.

Redirect the output to a file with the `-o` and `-O` options which will overwrite or append to an existing file, respectively. By default, the output is sent to `stdout`.

Use the `-r` option to set the default RIS reference type. This type will be used if `db2ris` cannot infer the reference type from the `BiblioEntry` element.

All other command line parameters will be interpreted as input filenames.

The following example shows how `db2ris` processes DocBook input files.

```
~# db2ris -o foo.ris *.sgml
```

This command will convert all `.sgml` files in the current directory and write the output to `foo.ris`.

15.6.2. The `db2risrc` variables

Table 15-7. `db2risrc`

Variable	Default	Comment
logfile	<code>/var/log/db2ris.log</code>	The full path of a custom log file. This is used only if <code>logdest</code> is set appropriately.
logdest	0	The destination of the log information. 0 = print to <code>stderr</code> ; 2 = use a custom logfile. The latter needs a proper setting of logfile.
authorlong	f	Set this to t if full first- and othernames should be used if possible.
defaultreftype	GEN	The default RIS reference type will be used if a <code>BiblioEntry</code> element does not specify the type.

15.6.3. `db2ris`' data mangling

Again we have to make a bunch of assumptions to arrive at a suitable output:

- db2ris will handle only “raw” (as opposed to “cooked”) bibliographic data. This basically means that you must encode all bibliographic data in `BiblioEntry` elements, not in `BiblioMixed` elements.
- db2ris discards all publication date information from the `PubDate` except the first 4 characters of `#PCDATA` so these should better be a 4-digit year.
- By default, the publication type is neither explicitly encoded in a `BiblioEntry` element nor can it be easily inferred. Therefore db2ris assumes that the `BiblioEntry` carry a `role` attribute with the RIS publication type as the value. If the attribute is missing, a default type is used instead. This means that you should add these attributes before the conversion in order to get useful RIS datasets (or fix the TY entries in the RIS file afterwards).
- DocBook does not have a special element to encode the name of a journal or magazine that published an article. Instead the same `Title` element is used as for the article title, a book title, or a series title. To distinguish between a journal or magazine name and “real” titles, the `Title` element encoding the journal name is assumed to be nested in a `BiblioSet` whose `relation` attribute is set to “journal”, “pub”, “abbrev”, or “full” (this more or less arbitrary list can of course be extended or changed, see below). The values “abbrev” and “full” furthermore decide whether the RIS tag “JO” or “JF” will be used.

15.6.4. Modifying db2ris

As previously stated, db2ris is implemented as a DSSSL stylesheet that can be easily and quickly adapted to your needs. It may be prudent to create copies of the script and modify these in order to adapt them for particular “abuses” of the DocBook bibliographic elements in various documents or data sources. This section briefly lists the functions which you would most likely want to change.

reftype-heuristic

This function attempts to infer the correct RIS publication type.

titletype-heuristic

This function gives a hint whether a `Title` element is used as a journal/magazine name or as an article title or book title.

process-date

This function creates a valid string for the RIS PY tag from the `PubDate` element.

15.7. marc2ris

This Perl script attempts to extract the information useful to reldb from MARC datasets. MARC (Machine Readable Catalogue Format) is a standard originating from the 1960s and is widely used by libraries and bibliographic agencies. Most libraries that offer Z39.50 access can provide the records in at

least one MARC format (like with most other "standards" there's a couple to choose from). Currently the following MARC dialects are supported:

MARC21

This is an attempt to consolidate existing MARC variants (mainly USMARC and CANMARC) and will most likely be the format supported by all libraries in the near future. The format is described on the Library of Congress MARC pages (<http://www.loc.gov/marc/>).

UNIMARC

This is the European equivalent of a standardization attempt. The specification can be found here (<http://www.ifla.org/VI/3/p1996-1/sec-uni.htm>).

UKMARC

This format is fairly close to the USMARC variant and is mainly used by libraries in the United Kingdom and in Ireland. Libraries supporting this format may switch to MARC21 in the future. Unfortunately there is no online description of this format, but this PDF document (www.bl.uk/services/bibliographic/marcchange.pdf) describes the main differences between USMARC and UKMARC.

15.7.1. Starting marc2ris

There's a variety of ways to run this script. By default the script reads USMARC data from stdin and sends RIS data to stdout, like this:

```
~$ perl marc2ris < foo.marc | less
```

Note: You can save some typing if (on Unix) the first line of the script points correctly to your Perl interpreter or if (on Windows) the filename suffix `.pl` is associated with the Perl interpreter. The following examples use this shorter invocation.

Alternatively you can specify one or more input files as arguments. Instead of displaying the results with a pager like in the previous command, we'll send the output to a file this time:

```
~$ marc2ris foo.marc bar.marc > foobar.ris
```

In either case you can specify an output file instead of sending the data to stdout. The following command will do exactly the same as the previous one:

```
~$ marc2ris -o foobar.ris foo.marc bar.marc
```

marc2ris accepts the following command line options and arguments:

```
marc2ris [-e log-destination] [-h] [-l log-level] [-L log-file] [-m] [-o outfile] [-O outfile] [-t input_type] [-u] [file...]
```

The `-h` option displays a brief usage screen and exits.

The `-m` option switches on additional MARC output, see below for details.

The `-o` and `-O` options send the RIS output to a file instead of to stdout. The lowercase option will overwrite an existing file, whereas the uppercase option will append to an existing file with the specified name.

The `-t` option allows to specify the MARC input type. The default is MARC21. Other available types are UNIMARC and UKMARC.

Use the `-u` option to request Unicode output. **marc2ris** attempts to convert the input data into Unicode (unless the dataset explicitly states that it already uses Unicode). Use this option with care as some MARC variants do not state the character encoding explicitly.

Note: The conversion routine supplied by the MARC::Record module uses a character conversion table designed for USMARC. This may or may not work with other MARC variants.

15.7.2. marc2ris data mangling

The purpose of the MARC format is entirely different from the purpose of the RIS format, so you shouldn't be too surprised that the import of MARC data is somewhat rough at the edges. The filter apparently deals fine with quite a lot of datasets, but the following shortcomings are known (and more are likely to be discovered by the interested reader):

- Some fields, like 846, are currently ignored completely. This, of course, is bound to change.
- Author names specified in the natural order, i.e. something like First Middle Last, are not normalized due to the problems with multiple middle or last names. Author names in the inverse order, i.e. something like Last, First Middle, are normalized correctly in most cases. Handling of non-European names is a matter of trial and error.
- Character set handling is somewhat limited. Only the unaltered input character encoding or UTF-8 are available for the output data.

That said, there is still some hope. The `-m` command line option switches on additional MARC output. That is, the generated output will contain interspersed lines that show the contents of the original MARC fields used to generate the following RIS line or lines. For example, the following output snippet shows how **marc2ris** generated the author lines from the MARC input:

```
<marc>empty author field (100)
<marc>:Author(Ind1): 1
<marc>:Author($a): Ershov, A. P.
<marc>:Author($b):
```

```

<marc>:Author($c):
<marc>:Author(Ind1): 1
<marc>:Author($a): Knuth, Donald Ervin,
<marc>:Author($b):
<marc>:Author($c):
AU - Ershov,A.P.
AU - Knuth,Donald Ervin

```

If you feel `marc2ris` does not translate your data appropriately, the easiest way might be to use the `-m` switch and redirect the output into a file. Then you can analyze the situation and fix the RIS lines as you see fit. Finally you can strip the MARC lines off with a command like:

```
~$ grep -v "<marc>" < withmarc.ris > womarc.ris
```

15.7.3. The `marc2ris` configuration variables

Table 15-8. `marc2risrc`

Variable	Default	Comment
outfile	(none)	The default output file name.
outappend	t	Determines whether output is appended (t) to an existing file or overwrites (f) an existing file.
unmapped	t	If set to t, unknown tags in the input data will be output following a <unmapped> tag; the resulting data can be inspected and then be sent through <code>sed</code> to strip off these additional lines. If set to f, unknown tags will be gracefully ignored.
logfile	/var/log/med2ris.log	The full path of a custom log file. This is used only if logdest is set appropriately.
logdest	1	The destination of the log information. 0 = print to stderr; 1 = use the syslog facility; 2 = use a custom logfile. The latter needs a proper setting of logfile.

Variable	Default	Comment
loglevel	6	The log level up to which messages will be sent. A low setting (0) allows only the most important messages, a high setting (7) allows all messages including debug messages. -1 means nothing will be logged.

15.8. Convert SGML and XML data to risx

Since version 0.9.3, reldb supports risx as an additional native input format. You can employ the standard techniques of SGML and XML transformations, i.e. by running DSSSL or XSLT scripts with a suitable engine, to turn bibliographic data encoded as SGML or XML documents into risx.

Remember that reldb accepts only four character encodings for XML input data: UTF-8, UTF-16, ISO-8859-1, and US-ASCII. If your input data use a different character encoding, please use the command-line utility **iconv** (usually part of the libiconv package) to convert your data to one of the existing character encodings. Do not forget to specify the character encoding in the processing instructions of the input file, otherwise reldb will assume the data are encoded as UTF-8.

Chapter 16. Bibliography tools

This chapter describes the helper tools that simplify the toil of creating bibliographies.

Note: As with all shell scripts that ship with `refdb`, the scripts described below have a user-customizable section right at the top of each file. You have to provide a few paths or set default values. If one of the scripts behaves strangely, please check again whether these values were correctly adapted to your local installation.

16.1. The `refdbnd` shell script

This script should be the first choice for novices to create new SGML or XML documents for use with `refdb`. If called without arguments, the script runs in an interactive mode and collects a few answers about the new document. Based on these answers it will create a skeleton document and a custom-tailored Makefile that performs all necessary steps to create formatted output from the document.

Alternatively you can call this script from a directory that contains an existing SGML or XML file. Pass the full name to the script when it asks for a filename, and the script will try to guess some of the settings from the existing file.

The script can create the following document types:

- DocBook SGML versions 3.1, 4.0, 4.1
- DocBook XML version 4.1.2
- TEI XML P4

The skeleton documents created by the script contain the appropriate prolog for the selected document type. The only extension is the declaration of an external entity in the internal subset which is required to include the bibliography.

To run the script in interactive mode, simply type the following command in an empty directory:

refdbnd

If you get bored by the verbose explanations you can specify all required arguments on the command line instead:

refdbnd {basename} {doctype} {pubtype} {database} {style} {encoding} {css-file}

The script will create two files in your current working directory:

- `Makefile`
- `<basename>.short.[sgml|xml]`

The Makefile is set up to process your document properly depending on its type. See the section about the `refdbnd` shortcut for a list of available targets.

16.2. The `runbib` shell script

This shell script is a wrapper for the bibliography client `refdbib` to simplify the creation of bibliographies. For SGML and XML documents, **runbib** first runs **openjade** or **xsltproc**, respectively, to retrieve a list of citations from your source document. Then it runs `refdbib` to generate the bibliography as an external entity based on the extracted citation information. It will also create stylesheet driver files with the formatting information for subsequent document transformations. For BibTeX documents, **runbib** uses the information in the `.aux` file to retrieve a bibliography file that you can use as an input file for **bibtex**.

```
runbib [-d database] [-E encoding] [-G css-file] [-h] [-i server-ip...] [-I] [-m
includevar...] [-N number] [-S style] [-t type] [-u username] [-w password] {file...}
```

The command-line switches are:

-d *database*

Select the default database. This database is used to retrieve the references unless the reference elements specify a database.

-E *encoding*

Specify the character encoding that the retrieved bibliography data should use.

-G *css-file*

Set the path or the URL of a CSS file to be used by the (x)html output

-h

Displays a help message and exits.

-i *server-ip*

Set the server IP address or hostname.

-I

Skip the ID extraction step. You'll have to create an appropriate `citationlistx` file yourself.

-m *includevar*

Specify variable names to include marked sections in the source document.

-N *number*

Start numbering the bibliography entries at *number*. This option applies only to numeric citation styles.

-S style

Select a bibliography style. This style must have been made available with the `refdba` command **addstyle**. To see which styles are available, use the **liststyle** command of `refdba` and `refdbc`.

-t type

Select the output type. This is one of “db31” (DocBook SGML documents), “db31x” (DocBook XML documents), “teix” (TEI XML documents), or “bibtex” (Latex documents).

All other command-line arguments are interpreted as input filenames.

Note: In the spirit of the LaTeX/BibTeX tools you can omit the `.aux` suffix in the input filename(s) for BibTeX output.

16.2.1. DocBook SGML output

`refdbib` generates two output files in addition to the ID listing created by Jade/OpenJade. `runbib` creates another output file from these. Assuming you have a source file `foo.sgml` and use the bibliography style `J.Biol.Chem.` you will get:

J.Biol.Chem.dsl

This is the DSSSL driver file file that encodes some additional formatting information specific for the chosen bibliography style for use in subsequent document transformations. This file will be written to the present working directory. The driver file is a dual-purpose stylesheet for HTML and printable output.

foo.id.xml

This is the intermediate XML file that encodes the citations in `foo.sgml`.

foo.bib.sgml

This is the file that holds the DocBook `bibliography` element. You have to include this file as an external entity into your source document to integrate the bibliography into your text.

J.Biol.Chem.css

This is a CSS file which contains formatting information used by the html output.

16.2.2. DocBook and TEI XML output

`refdbib` again generates two output files as for the DocBook SGML output above, and we also count the ID listing created by `xsltproc`. However, `runbib` does a little post-processing, so you’ll get two additional files in the end. Assuming you have a source file `foo.xml` and use the bibliography style `J.Biol.Chem.` you will get:

J.Biol.Chem.fo.xml

This is the XSL driver file used for printable output, containing additional formatting information specific for the chosen bibliography style. This file will be written to the present working directory.

J.Biol.Chem.html.xml

This is the XSL driver file used for HTML output, containing additional formatting information specific for the chosen bibliography style. This file will be written to the present working directory.

foo.id.xml

This is the intermediate XML file that encodes the citations in foo.xml.

foo.bib.xml

This is the file that holds the DocBook or TEI bibliography. You have to include this file as an external entity into your source document to integrate the bibliography into your text.

J.Biol.Chem.css

This is a CSS file which contains formatting information used by the html or xhtml output.

16.2.3. BibTeX output

refdbib creates one output file with the reference information. If your document is named `foo.tex`, **latex** will create a file `foo.aux` which refdbib uses as an input file. The refdbib output file will then be named `foo.bib` and can be used by **bibtex** as an input file.

16.2.4. Example

You should run this script from the directory where your document files are stored, as all output will be written to the present working directory by default.

The following command will create a DocBook SGML bibliography file together with a DSSSL stylesheet driver file for the DocBook SGML document `mypaper.sgml`. The output files are tweaked to match the citation and bibliography style of the "Journal of Biological Chemistry", which was added to the database with the name "J.Biol.Chem." previously. All references which do not contain a hard-coded database name are assumed to be in the database `bar`.

```
~$ runbib -d bar -S "J.Biol.Chem." -t db31 mypaper.sgml
```

16.3. The refdbjade shell script

This shell script uses the stylesheet driver file that you created with the `runbib` script and feeds it to Jade/OpenJade to transform your DocBook SGML or XML document to one of the supported output formats.

Note: If you prefer to transform your DocBook XML document with the XSL stylesheets as most people do these days, please use the `refdbxml` script.

16.3.1. Starting `refdbjade`

```
refdbjade [-h] [-i variable] [-p prefix] [-s stylesheet] [-t format] {file...}
```

The command-line switches are:

-h

Prints a command synopsis on the screen and exits

-i *variable*

Define a variable that is passed to Jade/OpenJade. Multiple entries are possible. This can be used to conditionally include or exclude parts of the document according to which variable is set.

-p *prefix*

This determines a prefix that is added to all filenames that Jade/OpenJade create. This can be useful to avoid filename conflicts if you transform or use several DocBook documents in the same folder.

-s *stylesheet*

This selects the stylesheet driver file. This file is generated by `refdbib` (which in turn is called by `runbib`) and contains additional formatting information.

-t *format*

Select an output format with this option. Possible values are "html", "rtf", "dvi", "pdf", "ps", "tex", to generate HTML, RTF, DVI, PDF, PostScript, or JadeTeX output, respectively. Use "tps" and "tpdf" to create PostScript or PDF output, respectively, from the intermediate JadeTeX output generated by any of the switches "dvi", "pdf", "ps", or "tex". This is useful if you have to edit the intermediate JadeTeX file, e.g. to manually shift pagebreaks. If your DSSSL engine of choice is OpenJade, you can use "htmlr" to generate raw HTML output (the same as HTML but without these funny linefeeds in the end tags).

16.3.2. Resolution of public identifiers

Public identifiers are resolved according to the SGML catalog files specified in `$SGML_CATALOG_FILES`. While **(open)jade** can retrieve DTDs from the web, it is advisable to keep local copies and add **VERRIDE YES** to the top of your catalog files. This ensures that the local copies are used.

16.3.3. Examples

Here we'll use the files generated in the last example above (see `runbib`) and generate a nicely formatted PDF file:

```
~$ refdbjade -d J.Biol.Chem.dsl -t pdf mypaper.sgml
```

To obtain HTML output, you just change the output type switch:

```
~$ refdbjade -d J.Biol.Chem.dsl -t html mypaper.sgml
```

16.4. The redbxml shell script

This is the XSL equivalent of the `refdbjade` script. Use it to transform your DocBook XML or TEI XML documents to HTML, PDF, or RTF output.

16.4.1. Starting redbxml

```
refdbxml [-c fop_config_file] [-f fo_processor] [-h] [-p xslt_processor] [-s  
stylesheet] [-t format] {file...}
```

The command-line switches are:

-h

Prints a command synopsis on the screen and exits

-p processor

This determines the XSL processor that is to be used. `refdbxml` currently knows to handle Xalan, XT, Saxon, and `xsltproc`. You can set your favourite processor as the default in the script so you don't have to use this option except when you need a little diversity. Of course you can only select a processor that is actually installed on your system.

-s stylesheet

Select the appropriate stylesheet driver file with this switch. The driver file that you want to use is one of the driver files which were generated by runbib. Use the HTML driver file for HTML output and the FO driver file for all other output. You can also use this script for non-RefDB documents. Use the values “db” and “tei” to select the plain DocBook and TEI XSL stylesheets, respectively.

-t format

Select an output format with this option. Possible values are html, rtf, and pdf, to generate HTML, RTF, and PDF output, respectively. The printable output formats depend on JFOR and PassiveTeX, respectively.

16.4.2. Configuring refdbxml

Instead of using the command-line switches, refdbxml can also be configured by means of the refdbxmlrc configuration file. As with all refdb configuration files, you may maintain a global copy in /usr/local/etc/refdb/ and one copy per user in \$HOME.

Table 16-1. refdbxmlrc

Variable	Default	Comment
xslt_processor	xsltproc	The name of the XSLT processor used to transform XML documents to html or fo files. Supported values are xsltproc (default), xalan, xt, saxon, saxon-xerces (using the xerces parser instead of the built-in parser)
xslt_classpath	/usr/share/java	Specify the directory which contains the Java classes for Java-based XSLT processors. This variable is not required if you use a non-Java processor (xsltproc).
fo_processor	fop	The name of the FO processor used to transform FO files into printable output. Supported values are fop (default), passivetex, xep, and xfor.

Variable	Default	Comment
fo_classpath	/usr/share/java	Specify the directory which contains the Java classes for Java-based FO processors. This variable is not required if you use a non-Java processor (passivetex).
fop_config_file	(none)	The path to a custom configuration file for FOP.
outformat	html	Set the default output format. Supported values are html, xhtml, pdf, and rtf. Be aware that pdf and rtf are not supported by all FO processors.

16.4.3. Resolution of public identifiers

Public identifiers can be resolved to local files if you have a working XML catalog on your system and if your XSLT processor supports XML catalogs. xalan and saxon require additional Java classes to support XML catalogs. For further information, please consult Bob Stayton's book (<http://www.sagehill.net/docbookxsl/UseCatalog.html>) about XSLT.

16.4.4. Examples

Again we'll transform our document (which now is either a DocBook or TEI XML document) to a nice-looking PDF file with the following command:

```
~$ refdbxml -d J.Biol.Chem.fo.xsl -t pdf mypaper.xml
```

Note that we used the FO stylesheet for this purpose. If we want HTML output, we need to change the output type switch and use the corresponding HTML stylesheet:

```
~$ refdbxml -d J.Biol.Chem.html.xsl -t html mypaper.xml
```

16.5. The bibdb2tei.xsl stylesheet

This XSLT stylesheet transforms a DocBook bibliography as generated by refdbib to a corresponding TEI bibliography for use in TEI XML documents. The runbib script performs this transformation automatically, but if you prefer to run refdbib directly you'll have to do this transformation yourself:


```
~$ xsltproc --catalogs /usr/local/share/refdb/xsl/bibdb2tei.xsl docbook.bib.xml > tei.bib.xml
```

where `docbook.bib.xml` is the bibliography file created by `refdbib`.

16.6. refdbxp

This application allows the interconversion of the short and full notation of citations in the supported SGML and XML documents. See the section about creating citations for a discussion of the two notations. The conversion works in both ways without a loss of information that couldn't be re-created by processing the file again the other way. This sounds a bit confusing but this statement is based on the following fact: while the full citation format contains the complete markup to transform the document with correctly formatted citations, the short format lacks two things:

- The markup for the individual references is missing. In the short format these are simple strings containing the ID or the citation key of a reference. However, the markup can easily be created from these strings.
- Citations in the short format do not distinguish between first and subsequent occurrences of a reference. However, as `refdbxp` scans the whole document it can derive this information from the text itself.

Therefore you can convert the citation style of your document back and forth as often as you wish. `refdbxp` also handles documents that contain both types of citations. This is convenient if you converted a document to the full citation style, but would like to add more citations in the more convenient short style. The only thing `refdbxp` cannot handle is if you try to mix both styles within the same citation, i.e. the following will not work:

```
<citation role="REFDB"><xref linkend="ID2X">5;</citation><!-- will not work -->
```

whereas the following will be handled gracefully:

```
<citation role="REFDB"><xref linkend="ID2X"></citation>
<!-- other stuff inbetween -->
<citation role="REFDB">2;5;</citation>
```

Note: You should be aware that `refdbxp` is not a SGML or XML-aware tool. It is a simple text replacement tool with some restrictions:

- If you comment out citation elements, they still count as if they were present when the first/subsequent citation issue is resolved (`refdbxp` simply doesn't know about the concept of a comment). In the following example, the citation in the last line will be the only one transformed, but it will be formatted as a subsequent citation of reference 9, not as the first citation:

```
<!-- <citation role="REFDB">9;</citation> first occurrence -->
<!-- other stuff inbetween -->
<citation role="REFDB">9;</citation><!-- second occurrence -->
```

- If you use SGML/XML tags within a comment and nest this comment ingeniously between the start tag and the end tag of an element relevant for `refdbxp`, you shoot yourself in the foot. You do not want to use code like this (why would you, anyways?):

```
<citation role="REFDB"><!-- </citation> -->2;5;9;</citation>
```

- `refdbxp` does not include external entities. The whole document `refdbxp` is supposed to convert needs to be in one chunk.
- `refdbxp` currently does not support multiple databases per document.

One way to work around the problem with comments is to create a copy of your master source and use a small script to remove comments just before you process and transform your text. To work around the fact that `refdbxp` does not treat external entities correctly, use a tool like **sgmlnorm** (shipped with the Jade/OpenJade packages) to preprocess the document. To work around the missing support of multiple databases, well... just wait.

16.6.1. Starting `refdbxp`

refdbxp [-h] [-s] [-t *input-format*]

Input is taken from stdin, output is written to stdout.

The command-line switches are:

-h

Prints a command synopsis on the screen and exits

-s

Create citations using the short notation. The default is to use the full notation.

-t *input-format*

Select the type of input. Currently supported values for *input-format* are `db31` (DocBook SGML version 3.1 or later), `db31x` (DocBook XML, all versions), and `teix` (TEI XML P3 or later).

Note: `refdbxp` might work with earlier versions of all DTDs but I didn't test that.

16.6.2. Examples

Lets first try the most common usage of `refdbxp`. The following command expands all citations, regardless of whether they are written in short or full notation, to the full notation and writes the result to a new file `foo.full.sgml`. The input from `foo.sgml` is assumed to be DocBook SGML:

```
~$ refdbxp -t db31 < foo.sgml > foo.full.sgml
```

The following command goes the other way. This time we convert all citations of a TEI XML document, regardless of whether they are written in short or full notation, to the short notation and write the result to a new file:

```
~$ refdbxp -t teix -s < bar.xml > bar.short.xml
```

The last example shows how to treat documents that consist of several files. The DocBook SGML master file `foo_master.sgml` includes several other subdocuments as external entities. Treating those files individually with `refdbxp` would screw up things as the first/subsequent citation issue would not be treated correctly and collisions of automatically created element IDs would result. The following command comes to the rescue and expands all citations in the document correctly:

```
~$ osgmlnorm -dn /usr/local/share/sgml/docbook/4.1/docbook.dcl foo_master.sgml | refdbxp -t db31 > fo
```

Note: You may have realized that there's two small problems with this procedure. First, using (o)sgmlnorm will also include the external entity that contains (or will contain once it's created) the bibliography element created by `refdb`. One way around this is to use a mock file that just contains the entity reference in a comment. Lets assume your document `foo.sgml` wants to include the bibliography by using the entity declaration `%bibliography;` at the proper location. The entity is declared in the declaration subset at the top of your sourcefile as the external file `foo.bib.sgml`. Then you should create a file `foo.bib.sgml` with the following contents:

```
<!--&bibliography;-->
```

We have to outcomment the entity reference as these may be nested, i.e. the parser would try to replace this entity again and fail because the entity is already opened. After the conversion you just need to uncomment the parameter entity. If you like long commands, you could do this on the fly like this:

```
~$ osgmlnorm -dn /usr/local/share/sgml/docbook/4.1/docbook.dcl foo_master.sgml | refdbxp -t db31 | sed 's%<
```

Second, (o)sgmlnorm will not output the internal declaration subset that we need at least to declare the parameter entity for the bibliography. You could fix this with a `sed` command along the lines of the command shown above or add it back manually.

V. Programmer's manual

Chapter 17. Reference database design

This section gives an overview over the design of the reference database with MySQL as the database server. The same design is used with the other database engines, although the details may differ somewhat. This will be mentioned where appropriate.

The requirements of the database necessitate some tables which hold the data proper and some tables which cross-reference these tables.

17.1. The table `t_meta`

The `t_meta` table contains the following meta-information about the database:

`meta_app`

This string contains the name of the application that created the database.

`meta_version`

This string contains the version number of the application that created the database.

`meta_type`

This string describes the type of the database. Currently only the type "risk" is supported.

`meta_create_date`

This timestamp is set by `refdbd` when the database is created.

`meta_modify_date`

This timestamp is updated by `refdbd` whenever the database is changed.

17.2. The table `t_refdb`

This table is the main table of the database and holds all fields which are unique to one reference entry and not different between users.

This table contains the following columns:

`refdb_id`

This unique identifier for entries in `t_refdb` is automatically generated by `refdb` and is stored as a `BIGINT` value (`INTEGER` for `SQLite`).

refdb_citekey

This is a user-supplied unique identifier for entries in t_refdb. If not supplied by the user, refdb will create a unique identifier when the entry is added.

refdb_type

This indicates the type of the document. This six-character string must be one of the following:

- ABST (abstract reference)
- ADVS (audiovisual material)
- ART (art work)
- BILL (bill/resolution)
- BOOK (whole book reference)
- CASE (case)
- CHAP (book chapter reference)
- COMP (computer program)
- CONF (conference proceeding)
- CTLG (catalog)
- DATA (data file)
- ELEC (electronic citation)
- GEN (generic)
- ICOMM (internet communication)
- INPR (in press reference)
- JFULL (journal - full)
- JOUR (journal reference)
- MAP (map)
- MGZN (magazine article)
- MPCT (motion picture)
- MUSIC (music score)
- NEWS (newspaper)
- PAMP (pamphlet)
- PAT (patent)
- PCOMM (personal communication)
- RPRT (report)
- SER (serial - book, monograph)
- SLIDE (slide)

- SOUND (sound recording)
- STAT (statute)
- THES (thesis/dissertation)
- UNBILL (unenacted bill/resolution)
- UNPB (unpublished work reference)
- VIDEO (video recording)

refdb_pubyear

This numerical value (SMALLINT) indicates the publication year.

refdb_secyear

This numerical value (SMALLINT) indicates the secondary year information.

refdb_startpage

This alphanumeric string (up to 255 characters) contains the start page information.

refdb_endpage

This alphanumeric string (up to 255 characters) contains the end page information.

refdb_abstract

This variable-length string contains the abstract or table of contents of the entry.

refdb_title

This alphanumeric field of variable length contains the title of the entry.

refdb_volume

This alphanumeric field (up to 255 characters) contains the volume number or identifier of the periodical.

refdb_issue

This alphanumeric field (up to 255 characters) contains the issue number or specifier.

refdb_booktitle

This alphanumeric field of variable length contains the book title (book chapter or whole book reference).

refdb_city

This alphanumeric field (up to 255 characters) contains the city where the periodical or book was published.

refdb_publisher

This alphanumeric field (up to 255 characters) contains the name of the publisher.

refdb_title_series

This alphanumerical field of variable length contains the title of a series of books or publications.

refdb_address

This alphanumeric field of variable length holds the address of the corresponding author and other contact information.

refdb_url

This alphanumeric field (up to 255 characters) holds a URL related to the entry, e.g. the homepage of an author or a link to an electronic reprint.

refdb_issn

This alphanumeric field with a maximum of 255 characters stores the ISSN or ISBN number of the publication.

refdb_pyother_info

This alphanumeric field with a maximum of 255 characters stores the additional information of a PY field after the publication year. The format is “/MM/DD/otherinfo”, with MM being the month and DD being the day of the publication. If either of these is missing, the corresponding slash “/” still has to be entered. The other information is free alphanumeric information.

refdb_secother_info

This alphanumeric field with a maximum of 255 characters provides the same additional information for the refdb_secyear field as the refdb_pyother_info field for refdb_pubyear.

refdb_periodical_id

This BIGINT (INTEGER for SQLite) variable points to the periodical_id in t_periodical which corresponds to the periodical the given article appeared in.

refdb_user1 through refdb_user5

These alphanumeric fields with a maximum of 255 characters provide space for user-defined information.

refdb_misc1 through refdb_misc3

These alphanumeric fields with a maximum of 255 characters provide space for user-defined information.

17.3. The table **t_author**

This table is a list of all authors, editors, and series editors.

author_id

The unique identifier of each author is stored as a BIGINT value (INTEGER for SQLite).

author_name

The name of the author is stored in an alphanumerical field (maximum length 255 characters). The name has the format Lastname[, (F. First) [(M. Middle) [, Suffix]]].

17.4. The table **t_keyword**

This table is a list of all keywords.

keyword_id

The unique identifier of each keyword is stored as a BIGINT value (INTEGER for SQLite).

keyword_name

This field holds the keyword (key phrase may be a better word as a keyword may consist of any alphanumeric string, including spaces). The maximum size is 255 characters.

17.5. The table **t_periodical**

This table is a list of all periodicals.

periodical_id

This is the unique identifier of each periodical and is stored as a BIGINT value (INTEGER for SQLite).

periodical_name

This is the full, unabbreviated name of the periodical. This is an alphanumeric field with a limit of 255 characters.

Examples: Trends in Biochemical Sciences; Proceedings of the National Academy of Sciences of the United States of America

periodical_abbrev

This is the official abbreviation of the periodical as seen in the *Index Medicus*. Abbreviated words have a trailing period. All words are separated by a space. This is also an alphanumeric field with a maximum of 255 characters.

Examples: Trends Biochem. Sci.; Proc. Natl. Acad. Sci. USA

periodical_custabbrev1

This is a custom abbreviation, often shorter and more popular than the official abbreviation. This is again an alphanumeric field with a limit of 255 characters.

Examples: TIBS; PNAS

periodical_custabbrev2

This is another custom abbreviation, often shorter and more popular than the official abbreviation. This is again an alphanumeric field with a limit of 255 characters.

17.6. The table **t_user**

This table contains the information about the users accessing the database. This information is used to track the person who added a specific article and the persons who are interested in a particular article.

user_id

The unique ID of each user is stored as a BIGINT value (INTEGER for SQLite).

user_name

This alphanumeric field holds the MySQL login name of the user (maximum length is 16 characters). The length limit is imposed by MySQL.

17.7. The table **t_xauthor**

This table cross-references the table **t_author** and **t_refdb**.

xauthor_id

This is the unique ID of a **t_xauthor** entry, stored as a BIGINT value (INTEGER for SQLite).

author_id

This is the ID of the author in the **t_author** table, stored as a BIGINT value (INTEGER for SQLite).

refdb_id

This is the ID of the reference in the `t_refdb` table, stored as a BIGINT value (INTEGER for SQLite).

xauthor_type

This ENUM field has the three possible values “primary”, “secondary”, and “tertiary” which denote that the person is a regular author, an editor, or a series editor in the given reference, respectively. PostgreSQL implements this as a SMALLINT value, SQLite uses a TEXT field.

xauthor_position

This INT field stores the original position of the author in the paper. The sequence of authors is taken from the sequence they appear in the RIS file.

17.8. The table `t_xkeyword`

This table cross-references the table `t_keyword` and `t_refdb`.

xkeyword_id

The unique ID of an `t_xkeyword` entry is stored as a BIGINT value (INTEGER for SQLite).

keyword_id

This BIGINT value (INTEGER for SQLite) points to the ID of the keyword in the table `t_keyword`.

refdb_id

This is the ID of the reference in the `t_refdb` table, stored as a BIGINT value (INTEGER for SQLite).

17.9. The table `t_xuser`

This table cross-references the table `t_user` and `t_refdb`.

xuser_id

The unique ID of an `t_xuser` entry is stored as a BIGINT value (INTEGER for SQLite).

user_id

This BIGINT value (INTEGER for SQLite) points to the ID of the user in the table `t_user`.

refdb_id

This is the ID of the reference in the t_refdb table, stored as a BIGINT value (INTEGER for SQLite).

xuser_reprint

This alphanumeric string must contain one of the following:

- NOT IN FILE (this is the default if nothing is specified)
- ON REQUEST
- IN FILE

xuser_date

This DATE field holds the date when a reprint was requested. If xuser_reprint contains something else than “ON REQUEST”, this field is not relevant.

xuser_avail

This alphanumeric field with a maximum of 255 characters contains the information where a physical copy of the article is stored. This may be a room number, a contact person, a binder or folder name or anything else that helps to track that copy down. This field is irrelevant if the reprint status is not “IN FILE”.

xuser_notes

This alphanumeric field of variable length contains notes or comments added by the user.

Chapter 18. The refdb database design

The refdb database is a helper database for the refdb reference databases. It holds data for the recognition of unabbreviated words in journal names as well as the bibliography style information.

18.1. The table `t_journal_words`

This table holds a list with unabbreviated journal words. These are words in the names of journals which are not an abbreviation of something else. Many online sources of bibliographic information provide the journal information without periods so it is not clear whether a word in a journal name is complete by itself or whether it is an abbreviation. Some bibliographic formats ask for periods after abbreviated words, so we have to get these periods from somewhere. refdb adds the periods if necessary when a reference is added to the database. This way you have the additional information that the periods provide always at your hands. If you don't need it, it is trivial to strip the periods away.

name

An alphanumeric field (maximum length 255 characters) that holds a word in uppercase which is an unabbreviated word in a journal name.

Examples: CELL, BIOCHEMISTRY, DRUGS

18.2. The table `CITSTYLE`

This table together with the `REFSTYLE`, `POSITION`, and `SEPARATORS` tables are used to store bibliography style sets. The `CITSTYLE` table contains one entry per bibliography style. It defines the appearance of the citations as well as the general appearance of the bibliography.

18.3. The table `REFSTYLE`

The `REFSTYLE` table contains zero or one entry for every publication type (such as book, journal, abstract) per bibliography style. Each entry is linked to one entry in the `CITSTYLE` table via the `CITSTYLEID` column.

18.4. The table `SEPARATORS`

The `SEPARATORS` table contains one entry per separator used in the `POSITION` table. Each entry is linked to an entry in the `REFSTYLE` table via the `REFSTYLEID` field.

18.5. The table POSITION

The SEPARATORS table contains one entry per separator used in the REFSTYLE table. Each entry is linked via the REFSTYLEID and POSITION fields.

Chapter 19. RIS and risx format specifications

In general, refdb tries to stick as closely as possible to the RIS format specification of the Reference Manager (<http://www.risinc.com>) software version 8 (this is the most recent version I was working with). This chapter briefly recalls the general syntax of RIS and risx files and the few differences in the implementation between Reference Manager and refdb.

19.1. The refdb RIS implementation

Import from Reference Manager and similar bibliographic databases as well as export should work in general, although it is not always loss-free due to the varying degree of support for the full RIS specification. This chapter briefly summarizes the differences between the Reference Manager RIS specification and what refdb uses.

- In contrast to the mainly Windows-based bibliographic databases, refdb expects the RIS files with UNIX line endings (LF = 0x0A). Most online sources allow to download the files in this format. The refdb package also contains a `dos2unix` shell script which converts the line endings from DOS format to UNIX format (you may use your favourite perl or whatever program instead, of course).
- In Reference Manager, the number of authors/editors/series authors per reference is limited to 255. Although this may not have any practical consequences, refdb does not have this restriction.
- The same holds true for keywords. In refdb the number of keywords per reference is not limited.
- Some of the tags defined in the RIS format specification have synonyms. refdb does not use these randomly upon output of RIS datasets, but uses a defined subset. This subset may be a different one than Reference Manager uses. This should have no practical consequences for the data integrity, although this means that after shuffling datasets between the two databases different synonymous tags may be used.
- The BT/T2 field is limited to 16KB in Reference Manager. It is limited to 64KB in refdb.
- A couple of fields in a Reference Manager database have either “no practical length limit” or the length limit is not specified at all. As these field lengths are not exactly given in SI units, I can just list what the corresponding limits in refdb are: PY/Y1, Y2 are limited to 255 characters starting from the first slash (i.e. excluding the year information), TI/T1/CT, IS/CP, BT/T2, T3, AD are limited to 64KB, AB and N1 are limited to 16MB. I assume that none of these poses a “practical length limit”. If you should plan to exceed these limits, you can of course recompile refdbd and use up to the maximum field length that the database server offers (e.g. 4GB in MySQL).

19.2. The risx DTD

The risx DTD allows to express RIS data in XML files. It has a few advantages over RIS, like full support for multiple-user notes, availability, and reprint information. The DTD is explained in a separate

manual (<http://refdb.sourceforge.net/risx/index.html>).

Chapter 20. The Perl client module

While the C clients shipped with refdb are quite versatile and scriptable, you may have a desire to bypass these clients and write scripts that directly talk to the refdb server. The separately available RefDBClient::Client Perl module implements the client/server communication necessary to run all commands offered by the C clients from a Perl script.

20.1. Installation

Like with most Perl modules, the following command sequence will install the Perl module on your system

```
~$ perl Makefile.PL  
  
~$ make  
  
~$ make install
```

Note: You need root permissions to run the last command on most systems.

20.2. Classes and their functions

Like most Perl modules, the RefDBClient::Client module is object-oriented. This section introduces the two classes that you need to know in order to work with the module. To see a working example, please check the `test.pl` script shipped with the module.

The main class that is used to access all refdb client functions is called RefDBClient::Client. To get started, create a new instance and set the communication parameters:

```
use RefDBClient::Client;  
  
my $client = new RefDBClient::Client;  
  
$client->set_conninfo("127.0.0.1", "9734", "markus", "pass", "refdbtest",  
"/home/markus/literature", "/usr/local/share/refdb/css/refdb.css");
```

Then you can go ahead and send commands to the server like this:

```
$summary = $client->refdb_listdb("");  
$data = $client->get_data();
```

Note: As you can easily guess the functions implementing the client commands are analogous to the commands found in the `refdba` and `refdbc` clients. See the reference pages about these apps for further information about the commands.

There are three helper classes:

- `RefDBClient::Risdata` provides a simple interface to RIS data
- `RefDBClient::Simplelist` is used internally
- `RefDBClient::Enigma` handles the password encryption

20.2.1. `RefDBClient::Risdata`

20.2.1.1. `new`

```
new RefDBClient::Risdata(void);
```

Creates a new `Risdata` object

20.2.1.2. `read_ris`

```
$data->read_ris($file);
```

loads RIS data from a file

`$file`

path of file

20.2.1.3. `get_ris`

```
$data->get_ris(void);
```

returns previously loaded RIS data

20.2.2. RefDBClient::Simplelist

20.2.2.1. new

```
new RefDBClient::Simplelist; (void);
```

creates a new Simplelist element

20.2.3. RefDBClient::Enigma

20.2.3.1. new

```
new RefDBClient::Enigma; (void);
```

creates a new Enigma element

20.2.4. RefDBClient::Client

20.2.4.1. new

```
new RefDBClient::Client; (void);
```

creates a new Client element

20.2.4.2. set_conninfo

```
$client->set_conninfo($server_ip, $port_address, $username, $password,  
$database, $pdf_root, $css_url);
```

sets the initial connection parameters of a Client object

\$server_ip

IP address or hostname of the server that runs refdbd

\$port_address

Port address at which refdbd listens

\$username

Username for database password authentication

\$password

Password for database password authentication

\$database

Name of the reference database

\$pdf_root

Path of the root directory of all electronic offprints

\$css_url

URL of a Cascading Stylesheets file for (X)HTML output

20.2.4.3. get_status

```
$client->get_status(void);
```

returns the numerical server status

20.2.4.4. get_status_msg

```
$client->get_status_msg(void);
```

returns the server status message

20.2.4.5. get_data

```
$client->get_data(void);
```

returns the data of the most recent command

20.2.4.6. **get_summary**

```
$client->get_summary(void);
```

returns the summary of the most recent command

20.2.4.7. **refdb_addstyle**

```
$client->refdb_addstyle($styledata);
```

adds a citation/bibliography style to the database

\$styledata

XML data representing the bibliography style

20.2.4.8. **refdb_adduser**

```
$client->refdb_adduser($host, $database, $newuserpassword, $username);
```

adds new users to the database

\$host

host specification from which the user is allowed to connect

\$database

name of the reference database

\$newuserpassword

password (required only for new users)

\$username

name of the user, as used to authenticate at the database engine

20.2.4.9. refdb_deleteuser

```
$client->refdb_deleteuser($host, $database, $username);
```

deletes users from the database

\$host

host specification from which the user is allowed to connect

\$database

name of the reference database

\$username

name of the user, as used to authenticate at the database engine

20.2.4.10. refdb_addword

```
$client->refdb_addword($words);
```

adds reserved words to the main database

\$words

space-separated list of words

20.2.4.11. refdb_deleteword

```
$summary = $client->refdb_deleteword($words);
```

removes reserved words from the main database

\$words

space-separated list of words

20.2.4.12. refdb_confserv

```
$client->refdb_confserv($command);
```

sends a configuration command to the server

\$command

the command proper, optionally followed by an argument

20.2.4.13. refdb_createdb

```
$client->refdb_createdb($dbname, $encoding);
```

creates a new database

\$dbname

name of the reference database

\$encoding

character encoding

20.2.4.14. refdb_deletedb

```
$client->refdb_deletedb($databasename);
```

deletes a reference database

\$dbname

name of the database

20.2.4.15. refdb_deletestyle

```
$client->refdb_deletestyle($stylename_regexp);
```

deletes citation/bibliography styles

\$stylename_regexp

regular expression describing the names of the styles to be deleted

20.2.4.16. refdb_getstyle

```
$client->refdb_getstyle($stylename);
```

retrieves a citation/bibliography style as a citestylex doc

\$stylename

name of the style

20.2.4.17. refdb_listdb

```
$client->refdb_listdb($dbname_regexp);
```

lists matching databases

\$dbname_regexp

regular expression describing the database names

20.2.4.18. refdb_listuser

```
$client->refdb_listuser($dbname, $username_regexp);
```

lists matching user names

\$username_regexp

regular expression describing the user names

20.2.4.19. refdb_listword

```
$client->refdb_listword($word_regexp);
```

lists matching journal name words

\$wordname_regexp

regular expression describing the word names

20.2.4.20. refdb_liststyle

```
$client->refdb_liststyle($stylename_regexp);
```

lists matching citation/bibliography styles

\$stylename_regexp

regular expression describing the style names

20.2.4.21. refdb_viewstat

```
$client->refdb_viewstat(void);
```

requests version/connection info from the server

20.2.4.22. refdb_sscanw

```
$client->refdb_sscanw($dbname);
```

runs a thorough keyword scan in the given database

\$dbname

name of the reference database

20.2.4.23. refdb_addrf

```
$client->refdb_addrf($owner, $refdata, $type, $encoding);
```

adds references to the database

\$owner

name of the dataset owner, if different from current user

\$refdata

string containing the reference data

\$type

data type, must be one of 'ris' or 'risk'

\$encoding

character encoding of the input data (only for RIS data)

20.2.4.24. refdb_updateref

```
$client->refdb_updateref($owner, $is_personal, $risdata, $type, $encoding);
```

updates references in the database

\$owner

name of the dataset owner, if different from current user

\$is_personal

set to 't' if only the personal information shall be updated

\$refdata

string containing the reference data

\$type

data type, must be one of 'ris' or 'risx'

\$encoding

character encoding of the input data (only for RIS data)

20.2.4.25. refdb_deleteref

```
$client->refdb_deleteref($idlist);
```

deletes references from the database

\$idlist

string specifying the IDs of the references to be deleted

20.2.4.26. refdb_addnote

```
$client->refdb_addnote($owner, $xnotedata);
```

adds notes to the database

\$owner

owner of the note, if different from the current user

\$xnotedata

XML data specifying the note

20.2.4.27. refdb_updatenote

```
$client->refdb_updatenote($owner, $is_personal, $xnotedata);
```

updates references in the database

\$owner

owner of the note, if different from the current user

\$is_personal

set to 't' if only the personal information shall be updated

\$xnotedata

XML data specifying the note

20.2.4.28. refdb_deletenote

```
$client->refdb_deletenote($idlist);
```

deletes notes from the database

\$idlist

string specifying the ID values of the notes to be deleted

20.2.4.29. refdb_addlink

```
$client->refdb_addlink($linkspec);
```

links notes to database objects

\$linkspec

string specifying the link(s) to be created

20.2.4.30. refdb_deletelink

```
$client->refdb_deletelink($linkspec);
```

unlinks notes from database objects

\$linkspec

string specifying the link(s) to be deleted

20.2.4.31. refdb_getas

```
$client->refdb_getas($name_regexp);
```

retrieves matching series authors

\$name_regexp

regular expression describing the names to be retrieved

20.2.4.32. refdb_getau

```
$client->refdb_getau($name_regexp);
```

retrieves matching authors

\$name_regexp

regular expression describing the names to be retrieved

20.2.4.33. refdb_geted

```
$client->refdb_geted($name_regexp);
```

retrieves matching editors

\$name_regexp

regular expression describing the names to be retrieved

20.2.4.34. `refdb_getkw`

```
$client->refdb_getkw($keyword_regexp);
```

retrieves matching keywords

`$keyword_regexp`

regular expression describing the keywords to be retrieved

20.2.4.35. `refdb_getjf`

```
$client->refdb_getjf($is_all, $journal_regexp);
```

retrieves matching periodicals (full names)

`$is_all`

set to 't' if all synonymous journal names shall be returned

`$name_regexp`

regular expression describing the names to be retrieved

20.2.4.36. `refdb_getjo`

```
$client->refdb_getjo($is_all, $journal_regexp);
```

retrieves matching periodical names (abbrev)

`$is_all`

set to 't' if all synonymous journal names shall be returned

`$name_regexp`

regular expression describing the names to be retrieved

20.2.4.37. refdb_getj1

```
$client->refdb_getj1($is_all, $journal_regexp);
```

retrieves matching periodical names (custom abbrev 1)

\$is_all

set to 't' if all synonymous journal names shall be returned

\$name_regexp

regular expression describing the names to be retrieved

20.2.4.38. refdb_getj2

```
$client->refdb_getj2($is_all, $journal_regexp);
```

retrieves matching periodical names (custom abbrev 2)

\$is_all

set to 't' if all synonymous journal names shall be returned

\$name_regexp

regular expression describing the names to be retrieved

20.2.4.39. refdb_getref

```
$client->refdb_getref($type, $format_string, $sort_string, $is_personal,  
$encoding, $query_string);
```

retrieves references

\$type

select output format

\$format_string

specify additional fields to be retrieved

\$sort_string

specify sorting key

\$is_personal

set to 't' to retrieve only references in the current users personal interest list

\$encoding

the character encoding for the output data

\$query_string

the query that describes the datasets to be retrieved

20.2.4.40. refdb_pickref

```
$client->refdb_pickref($idlist);
```

adds references to the users personal reference list

\$idlist

specifies the ID values of the references to be picked

20.2.4.41. refdb_dumpref

```
$client->refdb_dumpref($idlist);
```

removes references from personal reference list

\$idlist

specifies the ID values of the references to be dumped

20.2.4.42. refdb_getnote

```
$client->refdb_getnote($type, $format_string, $sort_string, $encoding,  
$query_string);
```


retrieves references

\$type

select output format

\$format_string

specify additional fields to be retrieved

\$sort_string

specify sorting key

\$encoding

the character encoding for the output data

\$query_string

the query that describes the datasets to be retrieved

20.2.4.43. refdb_selectdb

```
$client->refdb_selectdb($dbname);
```

selects an existing database as the current database

\$dbname

name of the reference database

20.2.4.44. refdb_whichdb

```
$client->refdb_whichdb(void);
```

displays information about the current database

20.2.4.45. refdb_texbib

```
$client->refdb_texbib($style, $cite_data);
```

retrieves a bibliography in bibtex format based on citationlistx data

\$style

the name of the citation/bibliography style

\$cite_data

XML data describing the references

20.2.4.46. refdb_dbib

```
$client->refdb_dbib($type, $style, $encoding, $cite_data);
```

retrieves an XML/SGML bibliography based on citationlistx data

\$type

type of the bibliography output

\$style

name of the citation/bibliography style

\$encoding

character encoding of the output data

\$cite_data

XML data specifying the references

Chapter 21. Using the backend API to implement new output formats

While there is nothing like a runtime-plugin mechanism to add new output formats, the `refdb` code is sufficiently modularized to make the implementation of a new format a fairly easy task. You don't have to parse the query results directly, but you call wrapper functions instead. This also has the nice advantage that changes in the database design show up only in one place and thus are less likely to break the backend implementations.

We will first have a look at the API that `backend.c` provides. Then we'll have a look at those parts in the existing code that need to be modified in order to accept a new backend.

21.1. The backend API

In general, the backend API provides a `get_foo()` for every tag *foo* in the RIS specification. There are, however, two fundamentally different types of tag retrieval functions:

1. The simple retrievals pull out values from the main table (`t_refdb`) of the database with a single function call.
2. The compound retrievals need three functions: A `request_foo()` obtains an array of possible values. A `get_foo()` retrieves one or more of these values and can be used in a loop to retrieve all values. Finally, the `clean_request()` frees the allocated memory. Compound retrievals are used to get at values which are stored outside the main table, like authors or keywords.

The prototypes of these functions can be found in `backend.h`, and their use is shown in the existing backends `backend-scrn.c`, `backend-ris.c`, `backend-db31.c`, and `backend-bibtex.c`.

21.2. How to insert a new backend into `refdb`

Define a new type, expand the if-then-else statement, include include-file. {work in progress}

Chapter 22. The refdb SGML/XML input and output formats

refdb uses XML files to encode the information about the required references for a bibliography. The resulting bibliographic output can be used as an external entity in both XML and SGML DocBook files (the bibliography can be transformed to any other SGML or XML type with suitable stylesheets). The structures of these files will be briefly explained in this chapter.

22.1. The XML input format for bibliographies

refdb uses (Open)Jade (for SGML documents) or an XSLT processor (for XML documents) to extract a list of required references and their logical relationships (position in the document, single or multiple citations and such). This list is encoded in an XML document conforming to the CitationList XML DTD (<http://refdb.sourceforge.net/dtd/citationlistx.dtd>).

22.2. The XML input format for bibliographic styles

The formatting instructions specific to a publisher or a journal where your document is to be published are encoded in a document conforming to the CiteStyle XML DTD. To increase the speed of the bibliography creation the formatting information has to be imported into refdb and is stored in the refdb database. The CiteStyle XML DTD (<http://refdb.sourceforge.net/dtd/citestylex/citestylex.dtd>) is moderately complex and is documented in a separate manual (<http://refdb.sourceforge.net/citestylex/index.html>).

22.3. The XML input format for extended notes

Extended notes are encoded in XML documents conforming to the Xnote DTD (<http://refdb.sourceforge.net/dtd/xnote/xnote.dtd>).

22.4. Processing expectations for the refdb DocBook bibliography output

The accepted standard DSSSL and XSL stylesheets for DocBook by Norman Walsh as well as the XSL stylesheets for TEI by Sebastian Rahtz are not designed to handle the complexity of the bibliography formatting requirements of various journals and publishers. refdb strains the limits of SGML and the document types to supply the required structural information, but it takes customized stylesheets to turn this into proper formatting. The refdb package contains sets of DSSSL and XSL stylesheets that were

designed for this purpose. They are implemented as driver files for the standard stylesheets. If those stylesheets do not do exactly what you need, you will have to modify them or write new ones from scratch. This chapter briefly explains the design of the output that refdb generates and how this should be processed in your stylesheets.

Note: This section discusses the DocBook output. It is straightforward to transfer this to TEI output. See [here](#) for a description of the equivalent TEI elements and attributes.

The general principle of the refdb bibliography is straightforward: Each citation that you want to be treated as a refdb citation needs to have a `role` attribute with the value “REFDB”. Each citation defines at least one `xref` element. The value of the `linkend` attribute encodes the ID of the required reference in the database (if you need references in several databases, this attribute can additionally specify the database). refdb uses this information to generate a DocBook `bibliography` element. This contains an entry for each requested reference. These entries are labelled with ID attributes that match the `xref linkend` attributes in the text. Each refdb-generated reference entry defines a `xreflabel` attribute which holds the text that is to be displayed at the position of the corresponding `xref` elements.

This is all it takes for single and unique citations, i.e. with one `xref` element per `citation` element and only one occurrence throughout the text. Both multiple occurrences of the same citation in the text and multiple citations (more than one `xref` elements per `citation` element) make things a bit more difficult.

Some output formats require a different formatting for the first citation of a publication in the text and all subsequent citations of the same publication. The first citation is identical with the above mentioned default case. All following citations of the same publication need an additional `xref endterm` attribute which points to an additional `bibliomset` element which in turn contains the text to be displayed for subsequent citations. The `endterm` attribute has the same value as the `linkend` attribute except that the letter “S” (as in subsequent) is appended to the attribute.

The real trouble starts with multiple citations. The output format may require the sequence of the citations to be sorted, and in the case of a numerical citation style it may require sequences of consecutive citations to be displayed as ranges (e.g. “[5-7]” instead of “[5,6,7]”). This may break the links between the individual citations and the reference in the bibliography in the output document. refdb will create an additional `bibliomset` element for each multiple citation. This provides the sorted and formatted text that is to be displayed for the multiple citation. For this to work you have to define an additional `xref` element whose `linkend` attribute points to one of the references and whose `endterm` points to the additional `bibliomset` element. This arrangement allows the multiple citation to be displayed differently depending on the desired output format. If the output is generated for printout, only the additional `xref` element should be formatted. This ensures the correct formatting of the citation in the printout. In output formats that allow hyperlinks it may be preferable to format the `xref` elements that link to the references individually. This may be incorrect in terms of the citation style, but the functional links to the references in the bibliography may outweigh this disadvantage.

Chapter 23. The refdb client/server communication protocol

This chapter describes the communication protocol that the refdb clients and server use to talk to each other. Knowledge of this protocol is useful for programmers who want to write custom refdb clients.

Tip: The refdb project provides a Perl client module in addition to the C clients shipped with the refdb package proper. Due to the simplicity of the Perl language, this module is a good resource for programmers who want to implement clients in other programming languages.

23.1. Prerequisites

Custom clients or client libraries can be implemented in any programming language that can create a Unix socket connection to the server and send/receive byte sequences through the socket. As the transferred data are essentially plain text, the endianness of the computer as well as the internal representation of data types in the particular programming language are irrelevant.

23.2. Basic principles

refdbd is implemented as a forking server. The parent process waits to accept connections from the clients. If a valid connection request is detected, the server forks. The parent closes the connection and is ready to respond to further requests. The child processes the client request and terminates when done.

The communication between the client and the server is at least a two-stage process. In the first stage, the validity of the client request is checked, the protocol version is checked and the password encryption is initiated. In the second stage, the command proper is executed and the results are sent back to the client. The second stage may use several iterations of client/server messages in order to transfer larger amounts of data.

23.2.1. Message format

All data, that is client commands as well as the server-generated results, are sent as plain text. We have to distinguish between three types of messages:

Status messages

Status messages are unterminated three-byte sequences which encode the client or server status. They may precede terminated messages, but they can also appear alone. The three bytes are a text representation of the client or server status. E.g. the sequence "000" (that is, three times the ASCII

character 48 representing the digit 'zero') denotes an OK status. A listing of the status messages used by refdb is shown in the section Status Messages.

Terminated messages

Most of the data transfer between client and server uses terminated messages. Due to the support of multibyte character sets the string termination character is not a single '\0' character as in a C string, but a sequence of four consecutive '\0' bytes (sequences of up to three '\0' bytes may occur as part of multibyte Unicode characters). It is best to think of the messages as binary strings. A custom client has to terminate its messages to the server appropriately, and it has to scan the data sent back by the server for the terminating sequence.

Messages of a specified length

In a few cases, the client asks the server to provide a buffer of a certain size, and will subsequently transfer exactly the requested number of bytes. These messages are unterminated.

23.2.2. First stage

The purpose of the first stage is to check whether the client request makes sense at all, and if so, to initiate the exchange of the password encryption information.

23.2.2.1. Sanity and permission check

Unless told otherwise, refdbd accepts only local connections. If the client request stems from a computer with a different IP address than the computer that runs refdbd, the connection request is refused without any further attempt to talk to the client. From the client side the connection will simply time out. This "unfriendly" behaviour minimizes the risk of remote exploits.

If the client is allowed to connect, refdbd tries to read a terminated string from the client. If this string is too short or too long to represent the protocol number which the client is supposed to send, the connection request is refused by sending back an appropriate status message. The same occurs if the protocol version of the client is not supported by the server.

23.2.2.2. Password encryption

The refdb clients may have to transmit the database username and password in order to authenticate the user with the database server. In order to avoid sending the password across the network as plain text, the clients send encrypted versions, which the refdbd server decrypts again. Database engine client libraries usually employ their own means to encrypt the passwords when they are sent from refdbd to the database engine.

refdb uses a fairly simple password encryption. It is still too cumbersome to decrypt for bored script kiddies so it should serve its purpose. The encryption is somewhat modeled after the (in)famous ENIGMA boxes used by the bad guys in WWII. The basic idea is that refdbd sends a string to the client which encodes the (randomly generated) rotor sequence and positions for this particular connection. The

client uses this information to encode the password. The server can decrypt the password using the sequence and position information that it previously generated.

The rotor wirings are hardcoded. Any client or library has to use the same wirings as in `src/filename.c`. The encoding string has the format "ABC-DE-FG-HI", which stands for:

ABC

This three-digit string denotes the sequence of the wheels. ABC denote the wheels in slots 0, 1, and 2, respectively. The wheels are also numbered 0 through 2, and each wheel can be used only once. That is, all valid combinations are "012", "021", "102", "120", "210", "201". For example, the string "120" means that wheel 1 is inserted in slot 0, wheel 2 in slot 1, and wheel 0 in slot 2.

DE

The position of the wheel in slot 0. A two-digit integer equal to or larger than 0 and smaller than 94. For example, the string "05" means the wheel in slot 0 has to advance 5 positions.

FG

Same as DE, but for slot 1.

HI

Same as DE, but for slot 2.

The encryption and decryption itself is described in sufficient detail in the source file `src/enigma.c` or in the `Client.pm` file of the `RefDBClient` Perl module. If you cannot implement this mechanism in your favourite programming language, there is also an external program **eenc** as part of the *refdb* distribution which performs the encryption/decryption and sends the result to `stdout`. Use it like this:

```
~$ eenc -p ABC-DE-FG-HI password
```

Most programming languages allow to read the results from `stdout` into a variable and use it as the encrypted or decrypted string.

Note: There is no switch for encryption or decryption as the ENIGMA mechanism is symmetrical.

23.2.3. Second stage

As the first step of the second stage the client sends the command string proper. These strings are commands as you know them from your shell. They are roughly equivalent to the command strings that you type into the C clients. *refdbd* parses the command string internally with the `getopt` library, just as your shell does. After the command string is parsed, *refdbd* executes the command. The result is returned in at least two steps. With one exception, *refdbd* first sends back the command result, followed by a command summary. For example, the `listdb` command first sends back a newline-separated list of

database names (the command result proper), followed by the number of available databases (the command summary). These steps are explained in more detail in the following sections. First we'll have a look at the internal command syntax.

23.2.3.1. Command syntax

Each command consists of the command word proper, optionally followed by switches, options, and arguments. As the string is parsed by the getopt library, the sequence of options, commands, and arguments does not matter. The following synopsis shows all possible switches and options.

command [-a] [-d *database*] [-E *encoding*] [-G *URL*] [-H *host*] [-k] [-n] [-N *password*] [-o *number*] [-p] [-P] [-r] [-R *pdfroot*] [-s *format_string*] [-S *sort_string*] [-t *ref_format*] [-u *username*] [-U *name*] [-w *password*]

-a

The commands `getjo`, `getjf`, `getj1`, and `getj2` use this switch to request all journal names (short, full, user abbrev1, and user abbrev2) instead of only the one used to match the query.

-d *database*

This option sets the name of the database to be used with the current command.

-E *encoding*

This option sets the character encoding for the current command.

-G *URL*

The `getref` command uses this option to pass the URL of a CSS stylesheet to the server.

-H *host*

The `adduser` command uses this option to set the hostname or IP address of the user to be added.

-k

This switch tells the `addref` command to preserve the numerical ID of the datasets in the U5 field.

-n

This switch causes the `getbib` command to send a style specification before transmitting the bibliography data proper.

-N *password*

The `adduser` command uses this option to set the password of the new user.

-o *number*

This option is used to set a custom starting number for numeric bibliographies.

-p

This option is used with the `addref` command. The allowed values mean: 0=add entry, 1=update according to ID field, 2=update personal info only.

-P

If used with the `getref` command, only references in the personal interest list will be returned.

-r

This switch is used with several commands to turn an "add" operation into a "remove" operation.

-R *pdfroot*

This option tells the `getref` command the variant part of the link to an electronic offprint.

-s *format_string*

This option can have the values "ris" or "risx" and denotes the type of the references to be added with the `addref` command.

-S *sort_string*

The name of the bibliography style to be used with the `getbib` command.

-t *ref_format*

The output format of the references retrieved with the `getref` and `getbib` commands.

-u *username*

The username to be used to authenticate with the database engine.

-U *name*

The name of the user to be associated with a query.

-w *password*

The password to be used to authenticate with the database engine.

23.3. Commented abstract representation of the client/server protocol

This chapter tries to explain the client/server protocol of all currently supported reldb commands using an abstract representation which should be fairly easy to port to any real programming language.

23.3.1. First stage

step	client	server	message type	comment
1	3\0\0\0\0			the current protocol version, a terminated string of up to 16 characters. The whole string will be converted to an integer internally.
2		000ABC-DE-FG-HI\0\0\0\0	ok	the ok status message, followed by the randomly generated encryption string
			error	there is no server response in case of a refused remote connection
		102, 103, 104, or 801	error	an error status message, ending the dialog
3	000command\0\0\0\0		ok	the ok status message, followed by the command proper. See below for details.
	112		error	the error status code signalling a client error

23.3.2. Second stage

The second stage is the more interesting part of the protocol, as it is here where the commands differ. The following section briefly explain the inner workings of the commands and show an abstract representation of the protocol.

23.3.2.1. addlink

The internal API command **addlink** corresponds to refdbc: addlink.

step	client	server	message type	comment
1	000addlink [options] link-spec [link-spec...]\0\0\0\0		ok	the ok message status, followed by the terminated addlink command sting
2		000<data>\0\0\0\0	ok	the ok message status, followed by a terminated string containing the command result
		803	partial success, aborted after unrecoverable error	the error message status, followed by a terminated string containing the command result
		204, 417, 233, or 801	error	the error message status which terminates the session
3	000		ok	the ok message status
4		000<data>\0\0\0\0	ok	the ok message status, followed by a terminated string containing the command result summary
5	000		ok	the ok message status, ending the dialog

23.3.2.2. addnote

The internal API command **addnote** corresponds to refdbc: addnote.

step	client	server	message type	comment
1	000addnote [options] filename\0\0\0\0		ok	the ok message status, followed by the terminated addnote command sting

step	client	server	message type	comment
2		000	ok	the ok message status
		701	warning	the warning message status
		202, 204, or 801	error	the error message status which terminates the session
3	000 <bytes>\0\0\0\0		ok	the ok message status, followed by a terminated string denoting the number of bytes that the client wants to transmit
	402		ok	the message status saying that all data have been sent, ending the loop over all datasets
	404		ok	the message status saying that the current dataset has been sent completely
	401		error	the message status saying that there was a problem reading the input data, ending the session
4		000	ok	the ok message status
		801	error	the error message status, ending the session
5	<data>		ok	the previously announced number of data bytes

step	client	server	message type	comment
6		403	ok	the ok status for a successfully transmitted chunk of data
		400 <message>\0\0\0 or 404	error	the error status, followed by a terminated, server-generated error message specifying the error
		408 or 413	ok	the message status denoting success of the previous add or update action
		702, 801	error	the error status, followed by a terminated, server-generated error message specifying the error
7	see step 3		ok	repeat loop until end of data
	402		ok	the message status saying that all data have been sent, ending the loop over all datasets
	404		ok	the message status saying that the current dataset has been sent completely

step	client	server	message type	comment
8		403 <result message>\0\0\0\0	ok	the ok status signalling that a chunk has been added successfully, followed by the terminated server result message, finishing the dialog
9	000		ok	the ok message status, ending the dialog

23.3.2.3. addref

The addref command can add bibliographic data in various formats to the database. The client/server protocols differ slightly based on the type of data transferred. RIS data are transmitted one dataset at a time, whereas XML data are streamed in chunks of a defined size.

step	client	server	message type	comment
1	000addref -s ris [options]\0\0\0\0		ok	the ok message status, followed by the terminated addref command string
2		000	ok	the ok message status
		701	warning	the warning message status
		202, 401, or 801	error	the error message status which terminates the session
3	000 <bytes>\0\0\0\0		ok	the ok message status, followed by a terminated string denoting the number of bytes that the client wants to transmit

step	client	server	message type	comment
	402		ok	the message status saying that all data have been sent, ending the loop over all datasets
	401		error	the message status saying that there was a problem reading the input data, ending the session
4		000	ok	the ok message status
		801	error	the error message status, ending the session
5	<data>		ok	the previously announced number of data bytes
6		403	ok	the ok status for a successfully transmitted chunk of data
		400 <message> \0\0\0\0	error	the error status, followed by a terminated, server-generated error message specifying the error
		408 or 413	ok	the message status denoting success of the previous add or update action
		702, 801	error	the error status, followed by a terminated, server-generated error message specifying the error

step	client	server	message type	comment
7	see step 3		ok	repeat loop until end of data
	402		ok	the message status saying that all data have been sent, ending the loop over all datasets
	404		ok	the message status saying that the current dataset has been sent completely
8		403	ok	the ok status signalling that a chunk has been added successfully
9	000		ok	the ok message status
10		000>data<\0\0\0	ok	the ok message status, followed by a terminated string containing the command result data
11	000		ok	the ok message status
12		000<data>\0\0\0	ok	the ok message status, followed by a terminated string containing the command result summary
13	000		ok	the ok message status, ending the dialog

And now the slightly different protocol for risk data:

step	client	server	message type	comment
------	--------	--------	--------------	---------

step	client	server	message type	comment
1	000addrf -s risx [options]\0\0\0\0		ok	the ok message status, followed by the terminated addrf command sting
2		000	ok	the ok message status
		701	warning	the warning message status
		202, 204, or 801	error	the error message status which terminates the session
3	000<bytes>\0\0\0\0		ok	the ok message status, followed by a terminated string denoting the number of bytes that the client wants to transmit
	402		ok	the message status saying that all data have been sent, ending the loop over all datasets
	404		ok	the message status saying that the current dataset has been sent completely
	401		error	the message status saying that there was a problem reading the input data, ending the session
4		000	ok	the ok message status
		801	error	the error message status, ending the session

step	client	server	message type	comment
5	<data>		ok	the previously announced number of data bytes
6		403	ok	the ok status for a successfully transmitted chunk of data
		400 <message> \0 \0 or	error	the error status, followed by a terminated, server-generated error message specifying the error
		408 or 413	ok	the message status denoting success of the previous add or update action
		702, 801	error	the error status, followed by a terminated, server-generated error message specifying the error
7	see step 3		ok	repeat loop until end of data
	402		ok	the message status saying that all data have been sent, ending the loop over all datasets
	404		ok	the message status saying that the current dataset has been sent completely
8		403	ok	the ok status signalling that a chunk has been added successfully

step	client	server	message type	comment
9	000		ok	the ok message status
10		000<data>\0\0\0	ok	the ok message status, followed by a terminated string containing the command result data
11	000		ok	the ok message status
12		000<data>\0\0\0	ok	the ok message status, followed by a terminated string containing the command result summary
13	000		ok	the ok message status, ending the dialog

23.3.2.4. addstyle

The internal API command **addstyle** corresponds to refdb: addstyle.

step	client	server	message type	comment
1	000addstyle [options] filename\0\0\0		ok	the ok message status, followed by the terminated addstyle command sting
2		000	ok	the ok message status
		104, 105, 106, 107, 108, 202, 203, 204, 206, or 801	error	the error message status which terminates the session

step	client	server	message type	comment
3	000<bytes>\0\0\0		ok	the ok message status, followed by a terminated string denoting the number of bytes that the client wants to transmit
	402		ok	the message status saying that all data have been sent, ending the loop over all datasets
	404		ok	the message status saying that the current dataset has been sent completely
	401		error	the message status saying that there was a problem reading the input data, ending the session
4		000	ok	the ok message status
		801	error	the error message status, ending the session
5	<data>		ok	the previously announced number of data bytes
6		403	ok	the ok status for a successfully transmitted chunk of data
		400<message>\0\0\0	error	the error status, followed by a terminated, server-generated error message specifying the error

step	client	server	message type	comment
7	see step 3		ok	repeat loop until end of data
	402		ok	the message status saying that all data have been sent, ending the loop over all datasets
	404		ok	the message status saying that the current dataset has been sent completely
8		403 <result message>\0\0\0\0	ok	the ok status signalling that a chunk has been added successfully, followed by the terminated server result message, finishing the dialog
9	000		ok	the ok message status, ending the dialog

23.3.2.5. adduser

The internal API command **adduser** corresponds to refdb: adduser.

step	client	server	message type	comment
1	000adduser [options]\0\0\0\0		ok	the ok message status, followed by the terminated adduser command sting
2		000	ok	the ok message status
		202224 or 801	error	the error message status which terminates the session

step	client	server	message type	comment
3	000<names>\0\0\0		ok	the ok message status, followed by a terminated string containing a list of names
4		000<summary>\0\0\0	ok	the ok message status
	205801		error	the message status saying that there was a problem, ending the session
5	000		ok	the ok message status, ending the dialog

23.3.2.6. addword

The internal API command **addword** handles both the refdb: addword and the refdb: deleteword client commands.

step	client	server	message type	comment
1	000addword [options]\0\0\0		ok	the ok message status, followed by the terminated addword command string
2		000	ok	the ok message status
		202, or 801	error	the error message status which terminates the session
3	000<data>\0\0\0		ok	the ok message status, followed by a terminated string containing the word list
	112		error	the error message status which terminates the session

step	client	server	message type	comment
4		000<data>\0\0\0	ok	the ok message status, followed by a terminated string containing the command result
5	000		ok	the ok message status
6		000<data>\0\0\0	ok	the ok message status, followed by a terminated string containing the command result summary
7	000		ok	the ok message status, ending the dialog

23.3.2.7. confserv

The internal API command **confserv** corresponds to refdb: confserv.

step	client	server	message type	comment
1	000confserv command [argument]\0\0\0		ok	the ok message status, followed by the terminated confserv command sting
2		000<data>\0\0\0	ok	the ok message status, followed by a terminated string containing the command result
		202, 301, 801, 839, 840	error	the error message status which terminates the session
3	000		ok	the ok message status

step	client	server	message type	comment
4		000<data>\0\0\0	ok	the ok message status, followed by a terminated string containing the command result summary
5	000		ok	the ok message status, ending the dialog

23.3.2.8. createdb

The internal API command **createdb** corresponds to refdb: createdb.

step	client	server	message type	comment
1	000createdb [options]dbname\0\0\0		ok	the ok message status, followed by the terminated createdb command sting
2		000<data>\0\0\0	ok	the ok message status, followed by a terminated string containing the command result
		111, or 801	error	the error message status which terminates the session
3	000		ok	the ok message status
4		000<data>\0\0\0	ok	the ok message status, followed by a terminated string containing the command result summary
5	000		ok	the ok message status, ending the dialog

23.3.2.9. deletedb

The internal API command **deletedb** corresponds to reldb: deletedb.

step	client	server	message type	comment
1	000deletedb [options]\0\0\0\0		ok	the ok message status, followed by the terminated deletedb command sting
2		000<data>\0\0\0\0	ok	the ok message status, followed by a terminated string containing the command result
		202, or 801	error	the error message status which terminates the session
3	000		ok	the ok message status
4		000<data>\0\0\0\0	ok	the ok message status, followed by a terminated string containing the command result summary
5	000		ok	the ok message status, ending the dialog

23.3.2.10. deletenote

The internal API command **deletenote** corresponds to reldbc: deletenote.

step	client	server	message type	comment
------	--------	--------	--------------	---------

step	client	server	message type	comment
1	000deletenote [options] <bytes>\0\0\0\0		ok	the ok message status, followed by the terminated adduser command sting containing the number of bytes required to store the ID list.
2		000	ok	the ok message status
		111801	error	the error message status which terminates the session
3	000<ID-list>\0\0\0\0		ok	the ok message status, followed by a terminated string containing a list of note IDs
4		000<data>\0\0\0\0	ok	the ok message status, followed by a report about the delete actions
		204, 412, 801	error	the message status saying that there was a problem, ending the session
5	000		ok	the ok message status
6		000<summary>\0\0\0\0	ok	the ok message status, followed by a command summary
7	000		ok	the ok message status, ending the dialog

23.3.2.11. deleteref

The internal API command **deleteref** corresponds to refdbc: deleteref.

step	client	server	message type	comment
1	000deleteref [options] <bytes>\0\0\0\0		ok	the ok message status, followed by the terminated adduser command string containing the number of bytes required to store the ID list.
2		000	ok	the ok message status
		801	error	the error message status which terminates the session
3	000<ID-list>\0\0\0\0		ok	the ok message status, followed by a terminated string containing a list of reference IDs
4		000<data>\0\0\0\0	ok	the ok message status, followed by a report about the delete actions
		204, 412, 801	error	the message status saying that there was a problem, ending the session
5	000		ok	the ok message status
6		000<summary>\0\0\0\0	ok	the ok message status, followed by a command summary
7	000		ok	the ok message status, ending the dialog

23.3.2.12. deletestyle

The internal API command **deletestyle** corresponds to refdba: deletestyle.

step	client	server	message type	comment
1	000deletestyle {regex}\0\0\0\0		ok	the ok message status, followed by the terminated deletestyle command sting
2		000<data>\0\0\0\0	ok	the ok message status, followed by a terminated string containing the command result
		202, 234, or 801	error	the error message status which terminates the session
3	000		ok	the ok message status
4		000<data>\0\0\0\0	ok	the ok message status, followed by a terminated string containing the command result summary
5	000		ok	the ok message status, ending the dialog

23.3.2.13. getau, geted, getas, getkw, getjo, getjf, getj1, getj2

These internal API commands correspond to the client commands refdb: getau, refdb: geted, refdb: getas, refdb: getkw, refdb: getjo, refdb: getjf, refdb: getj1, refdb: getj2.

step	client	server	message type	comment
1	000getXX [options]\0\0\0\0		ok	the ok message status, followed by the terminated command sting
2		000<data>\0\0\0\0	ok	the ok message status, followed by a terminated string containing the command result

step	client	server	message type	comment
		204, 208, 234, or 801	error	the error message status which terminates the session
3	000		ok	the ok message status
4		000<data>\0\0\0	ok	the ok message status, followed by a terminated string containing the command result summary
5	000		ok	the ok message status, ending the dialog

23.3.2.14. getbib

The internal API command **getbib** is used by the refdbib tool to retrieve SGML and XML bibliographies.

step	client	server	message type	comment
1	000getbib [options]\0\0\0		ok	the ok message status, followed by the terminated getbib command string.
2		000	ok	the ok message status, meaning no style spec will be transmitted (jump to 4)
		402	ok	the ok message status, meaning the style spec will be transmitted
		202, 204, 241, 701, 801	error	the error message status, terminating the dialog
3	000		ok	the ok message status

step	client	server	message type	comment
4		000	ok	the ok message status
		242	error	the error message status, finishing the dialog
5	000 <bytes>\0\0\0\0		ok	the ok message status, followed by a terminated string denoting the number of bytes that the client wants to transmit
	402		ok	the message status saying that all data have been sent, ending the loop over all datasets
	404		ok	the message status saying that the current dataset has been sent completely
	401		error	the message status saying that there was a problem reading the input data, ending the session
6		000	ok	the ok message status
		801	error	the error message status, ending the session
7	<data>		ok	the previously announced number of data bytes
8		403	ok	the ok status for a successfully transmitted chunk of data

step	client	server	message type	comment
		400<message>\0\0\0	error	the error status, followed by a terminated, server-generated error message specifying the error
9	000		ok	the ok message status
10		404	ok	the ok status for a successfully transmitted chunk of data
		402	ok	the ok status for the last chunk of data
11	000		ok	the ok message status
12		000<summary>\0\0\0	ok	the ok message status, followed by the terminated command summary string.
11	000		ok	the ok message status, finishing the dialog

23.3.2.15. getnote

The internal API command **getnote** corresponds to refdb: getnote.

step	client	server	message type	comment
1	000getnote [options] search- string\0\0\0\0		ok	the ok message status, followed by the terminated getnote command string

step	client	server	message type	comment
2		404 <data>\0\0\0	ok	the ok message status denoting a complete dataset, followed by a terminated string containing the dataset
		402 <data>\0\0\0	ok	the ok message status denoting the last chunk of data (usually a "footer"), followed by a terminated string containing the data
		204, 234, 701, or 801	error	the error message status which terminates the session
3	000		ok	the ok message status
4		000 <data>\0\0\0	ok	the ok message status, followed by a terminated string containing the command result summary
4		402 <data>\0\0\0	ok	the ok message status denoting the last chunk of data (usually a "footer"), see step 2
5		000 <data>\0\0\0	ok	the ok message status, followed by a terminated string containing the command result summary

step	client	server	message type	comment
		702, or 801	error	the error message status which terminates the session
6	000		ok	the ok message status, ending the dialog

23.3.2.16. getref

The internal API command **getref** corresponds to refdbc: getref.

step	client	server	message type	comment
1	000getref [options] <bytes>\0\0\0		ok	the ok message status, followed by the terminated getref command sting containing the number of bytes required to store the query string
2		000	ok	the ok message status
		801	error	the error message status which terminates the session
3	000<query-string>\0\0\0		ok	the ok message status, followed by a terminated string containing the query
4		404<data>\0\0\0	ok	the ok message status denoting a complete dataset, followed by a terminated string containing the dataset

step	client	server	message type	comment
		402 <data>\0\0\0	ok	the ok message status denoting the last chunk of data (usually a "footer"), followed by a terminated string containing the data
		204, 234, 701, or 801	error	the error message status which terminates the session
5	000		ok	the ok message status
6		402 <data>\0\0\0	ok	the ok message status denoting the last chunk of data (usually a "footer"), see step 2
7		000 <data>\0\0\0	ok	the ok message status, followed by a terminated string containing the command result summary
		702, or 801	error	the error message status which terminates the session
8	000		ok	the ok message status, ending the dialog

23.3.2.17. getstyle

The internal API command **getstyle** corresponds to refdb: getstyle.

step	client	server	message type	comment
------	--------	--------	--------------	---------

step	client	server	message type	comment
1	000getstyle [options][regexp]\0\0\0		ok	the ok message status, followed by the terminated getstyle command string
2		000<data>\0\0\0	ok	the ok message status, followed by a terminated string containing the command result
		202, or 801	error	the error message status which terminates the session
3	000		ok	the ok message status, ending the dialog

23.3.2.18. gettexbib

The internal API command **gettexbib** is used by the redbib tool to retrieve bibtex bibliographies.

step	client	server	message type	comment
1	000gettexbib [options] data-size\0\0\0		ok	the ok message status, followed by the terminated gettexbib command string. The only non-option argument is the size, in bytes, of the ID data to be sent, including the terminator.
2		000	ok	the ok message status
		801	error	the error message status, terminating the dialog

step	client	server	message type	comment
3	000id-data\0\0\0\0		ok	the ok message status, followed by the terminated ID list. This is a whitespace separated list of ID or CK values.
4		404<data>\0\0\0\0	ok	the ok message status denoting a complete dataset, followed by a terminated string containing the dataset
		402<data>\0\0\0\0	ok	the ok message status denoting the last chunk of data (usually a "footer"), followed by a terminated string containing the data
		204, 234, 701, or 801	error	the error message status which terminates the session
5	000		ok	the ok message status
6		402<data>\0\0\0\0	ok	the ok message status denoting the last chunk of data (usually a "footer"), see step 2
7		000<data>\0\0\0\0	ok	the ok message status, followed by a terminated string containing the command result summary

step	client	server	message type	comment
		702, or 801	error	the error message status which terminates the session
8	000		ok	the ok message status, ending the dialog

23.3.2.19. listdb

The internal API command **listdb** corresponds to refdbc: listdb.

step	client	server	message type	comment
1	000listdb [regexp]\0\0\0		ok	the ok message status, followed by the terminated listdb/selectdb command sting
2		000<data>\0\0\0	ok	the ok message status, followed by a terminated string containing the command result
		202, 225, 226, 802, or 802	error	the error message status which terminates the session
3	000		ok	the ok message status
4		000<data>\0\0\0	ok	the ok message status, followed by a terminated string containing the command result summary
5	000		ok	the ok message status, ending the dialog

23.3.2.20. liststyle

The internal API command **liststyle** corresponds to refdb: liststyle.

step	client	server	message type	comment
1	000liststyle [options][regexp]\0\0\0		ok	the ok message status, followed by the terminated liststyle command sting
2		000<data>\0\0\0	ok	the ok message status, followed by a terminated string containing the command result
		202, or 801	error	the error message status which terminates the session
3	000		ok	the ok message status
4		000<data>\0\0\0	ok	the ok message status, followed by a terminated string containing the command result summary
5	000		ok	the ok message status, ending the dialog

23.3.2.21. listuser

The internal API command **listuser** corresponds to refdb: listuser.

step	client	server	message type	comment
1	000listuser [options] regexp\0\0\0		ok	the ok message status, followed by the terminated listuser command sting

step	client	server	message type	comment
2		000<data>\0\0\0\0	ok	the ok message status, followed by a terminated string containing the command result
		204, 207, or 801	error	the error message status which terminates the session
3	000		ok	the ok message status
4		000<data>\0\0\0\0	ok	the ok message status, followed by a terminated string containing the command result summary
5	000		ok	the ok message status, ending the dialog

23.3.2.22. listword

The internal API command **listword** corresponds to refdb: listword.

step	client	server	message type	comment
1	000listword regexp\0\0\0\0		ok	the ok message status, followed by the terminated listword command sting
2		000<data>\0\0\0\0	ok	the ok message status, followed by a terminated string containing the command result
		204, 207, or 801	error	the error message status which terminates the session

step	client	server	message type	comment
3	000		ok	the ok message status
4		000<data>\0\0\0	ok	the ok message status, followed by a terminated string containing the command result summary
5	000		ok	the ok message status, ending the dialog

23.3.2.23. pickref

The internal API command **pickref** handles the client commands refdb: pickref and dumpref.

step	client	server	message type	comment
1	000pickref [options] <bytes>\0\0\0		ok	the ok message status, followed by the terminated pickref command sting containing the number of bytes required to store the ID list.
2		000	ok	the ok message status
		111801	error	the error message status which terminates the session
3	000<ID-list>\0\0\0		ok	the ok message status, followed by a terminated string containing a list of note IDs
4		000<data>\0\0\0	ok	the ok message status, followed by a report about the pick or dump actions

step	client	server	message type	comment
		204, 412, 801	error	the message status saying that there was a problem, ending the session
5	000		ok	the ok message status
6		000<summary>\0\0\0	ok	the ok message status, followed by a command summary
		227, 228, 234	error	the error status, terminating the dialog
7	000		ok	the ok message status, ending the dialog

23.3.2.24. scankw

The internal API command **scankw** corresponds to refdb: scankw.

step	client	server	message type	comment
1	000scankw -d <dbname>\0\0\0		ok	the ok message status, followed by the terminated scankw command sting
2		000<data>\0\0\0	ok	the ok message status, followed by a terminated string containing the command result
		204, 208, or 801	error	the error message status which terminates the session
3	000		ok	the ok message status

step	client	server	message type	comment
4		000<data>\0\0\0	ok	the ok message status, followed by a terminated string containing the command result summary
5	000		ok	the ok message status, ending the dialog

23.3.2.25. selectdb

The internal API command **selectdb** corresponds to refdbc: selectdb.

step	client	server	message type	comment
1	000selectdb dbname\0\0\0\0		ok	the ok message status, followed by the terminated listdb/selectdb command sting
2		000<data>\0\0\0	ok	the ok message status, followed by a terminated string containing the command result
		202, 225, 226, 802, or 802	error	the error message status which terminates the session
3	000		ok	the ok message status
4		000<data>\0\0\0	ok	the ok message status, followed by a terminated string containing the command result summary
5	000		ok	the ok message status, ending the dialog

23.3.2.26. viewstat

The internal API command **viewstat** corresponds to refdba: viewstat.

step	client	server	message type	comment
1	000viewstat [options]\0\0\0\0		ok	the ok message status, followed by the terminated viewstat command sting
2		000<data>\0\0\0\0	ok	the ok message status, followed by a terminated string containing the command result
		204, 208, or 801	error	the error message status which terminates the session
3	000		ok	the ok message status
4		000<data>\0\0\0\0	ok	the ok message status, followed by a terminated string containing the command result summary
5	000		ok	the ok message status, ending the dialog

23.3.2.27. whichdb

The internal API command **whichdb** corresponds to refdbc: whichdb.

step	client	server	message type	comment
1	000whichdb\0\0\0\0		ok	the ok message status, followed by the terminated whichdb command sting

step	client	server	message type	comment
2		000<data>\0\0\0	ok	the ok message status, followed by a terminated string containing the command result
		204, 207, or 801	error	the error message status which terminates the session
3	000		ok	the ok message status
4		000<data>\0\0\0	ok	the ok message status, followed by a terminated string containing the command result summary
5	000		ok	the ok message status, ending the dialog

23.3.3. Status messages

Note: The status codes which are multiples of 100 are not associated with a fixed message. Instead, clients should retrieve the terminated string sent after these codes to read the dynamically generated message.

000

ok

001

error

100

void

- 101**
 - incorrect scramble string
- 102**
 - client and server protocols do not match
- 103**
 - invalid client request
- 104**
 - incomplete client command
- 105**
 - missing client command
- 106**
 - missing client command option
- 107**
 - unknown client command option
- 108**
 - could not descramble password
- 109**
 - timeout while reading
- 110**
 - timeout while writing
- 111**
 - missing client command argument
- 112**
 - client aborted command
- 200**
 - void
- 201**
 - main database is missing

- 202**
could not open main database
- 203**
main database is too old or corrupt
- 204**
could not open reference database
- 205**
could not connect to database server
- 206**
main database version is not supported
- 207**
could not create result from database query
- 208**
could not retrieve reference database metadata
- 209**
could not create reference database
- 210**
could not create reference database metadata
- 211**
create t_meta failed
- 212**
create t_refdb failed
- 213**
create t_author failed
- 214**
create t_keyword failed
- 215**
create t_periodical failed

216

create t_note failed

217

create t_user failed

218

create t_xauthor failed

219

create t_xkeyword failed

220

create t_xuser failed

221

create t_xnote failed

222

could not create user group

223

could not grant user permissions

224

access control not supported

225

not a RefDB database

226

database does not exist

227

begin transaction failed

228

cannot lock tables

229

failed to remove keyword

- 230**
 - failed to remove author
- 231**
 - failed to remove periodical
- 232**
 - failed to update main reference data
- 233**
 - inserting reference data failed
- 234**
 - select failed
- 235**
 - database successfully created
- 236**
 - assume localhost as host
- 237**
 - grant user permissions successful
- 238**
 - revoke user permissions successful
- 239**
 - could not revoke user permissions
- 240**
 - switched to database
- 241**
 - failed to access style data
- 242**
 - create temporary table failed
- 243**
 - delete temporary table failed

- 244**
 - incomplete reference data
- 245**
 - failed to remove note xlink
- 246**
 - failed to delete main note data
- 247**
 - failed to remove user
- 248**
 - failed to delete main reference data
- 249**
 - failed to delete database
- 250**
 - could not delete user group
- 251**
 - database successfully deleted
- 252**
 - personal interest list is empty
- 253**
 - failed to detach dataset from user
- 254**
 - sucessfully detached dataset from user
- 255**
 - failed to attach dataset to user
- 256**
 - sucessfully attached dataset to user
- 257**
 - create t_link failed

258	create t_xlink failed
259	failed to remove ulink
260	failed to update journal names
300	void
301	missing argument
302	unknown output format
400	void
401	no more data available
402	finished transferring data
403	chunk added successfully
404	finished transferring dataset
405	finished adding dataset
406	citation key
407	refused to overwrite dataset

- 408**
 - dataset added successfully
- 409**
 - numerical id ignored
- 410**
 - numerical id nonexistent
- 411**
 - citation key nonexistent
- 412**
 - ID and citation key missing
- 413**
 - dataset updated successfully
- 414**
 - failed to add dataset
- 415**
 - missing link target
- 416**
 - incorrect link type
- 417**
 - dataset not found
- 418**
 - link already exists
- 419**
 - dataset removed successfully
- 420**
 - failed to remove dataset
- 421**
 - link added successfully

422	only owner can fiddle with dataset
423	dataset is still in use
424	dataset is already attached to user
425	periodical name changed successfully
700	void
701	failed to initialize character set conversion
702	character set conversion failed
703	remote administration disabled
704	administration is not restricted
705	administration not permitted
706	administration permitted
707	process ID
708	application server stop submitted
709	set server IP submitted

- 710**
 - set timeout submitted
- 711**
 - set logfile submitted
- 712**
 - set logdest submitted
- 713**
 - set loglevel submitted
- 800**
 - void
- 801**
 - out of memory
- 802**
 - failed to load cgi templates
- 803**
 - command partially processed, aborted after unrecoverable error
- 804**
 - suffix pool exhausted
- 805**
 - REFNUMBER formatting failed
- 806**
 - AUTHORLIST formatting failed
- 807**
 - EDITORLIST formatting failed
- 808**
 - SEEDITORLIST formatting failed
- 809**
 - PUBDATE formatting failed

- 810**
PUBDATESEC formatting failed
- 811**
TITLE formatting failed
- 812**
BOOKTITLE formatting failed
- 813**
SERIESTITLE formatting failed
- 814**
JOURNALNAME formatting failed
- 815**
VOLUME formatting failed
- 816**
ISSUE formatting failed
- 817**
PAGES formatting failed
- 818**
PUBLISHER formatting failed
- 819**
PUBPLACE formatting failed
- 820**
SERIAL formatting failed
- 821**
ADDRESS formatting failed
- 822**
URL formatting failed
- 823**
USERDEF1 formatting failed

824
USERDEF2 formatting failed

825
USERDEF3 formatting failed

826
USERDEF4 formatting failed

827
USERDEF5 formatting failed

828
MISC1 formatting failed

829
MISC2 formatting failed

830
MISC3 formatting failed

831
LINK1 formatting failed

832
LINK2 formatting failed

833
LINK3 formatting failed

834
LINK4 formatting failed

835
ABSTRACT formatting failed

836
NOTES formatting failed

837
SEPARATOR formatting failed

838

remote administration failed

839

child->parent communication failure

840

FIFO write error

841

unknown command

999

summary

Colophon

This document was written with GNU Emacs (<http://www.gnu.org/software/emacs>) on various versions of Debian GNU/Linux (<http://www.debian.org>) and FreeBSD (<http://www.freebsd.org>). The SGML (<http://xml.coverpages.org/sgml.html>) sources use the DocBook SGML DTD (<http://www.docbook.org/sgml/4.1/index.html>) version 4.1. The images were originally created with OpenOffice Draw (<http://www.openoffice.org/product2/draw.html>). They were exported as SVG (<http://www.w3.org/TR/SVG/>) vector graphics and are now being maintained using Inkscape (<http://www.inkscape.org/>). The SVG files were transformed to PNG (<http://www.w3.org/Graphics/PNG/>) and PDF (<http://www.adobe.com/products/acrobat/adobepdf.html>) using the Java (<http://java.sun.com/>) application batik-rasterizer (<http://xml.apache.org/batik/>).

The document was transformed to the PDF and HTML (<http://www.w3.org/Markup/>) output formats using OpenJade (<http://openjade.sourceforge.net/>) and the DocBook DSSSL stylesheets (<http://docbook.sourceforge.net/projects/dsssl/>) through customization layers which are part of the documentation sources. The HTML output is further enhanced by a CSS (<http://www.w3.org/Style/CSS/>) file.