RefDB handbook

covers version 0.9.9

Markus Hoenicka

RefDB handbook: covers version 0.9.9

Markus Hoenicka

Table of Contents

List of Figures

List of Tables

Preface

1. RefDB in a nutshell

RefDB is a free and portable reference and notes database toolkit 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. There is a choice of command line clients, web interfaces, editor interfaces, and language bindings that allow you to work with your reference data any way you want.

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 casual hackers or notorious code fiddlers.

3. Further information

This document is complemented by the RefDB tutorial [http://refdb.sourceforge.net/doc/tutorial/index.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 to browse the list archives.

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 to browse the list archives.

The refdb-cvs [mailto:refdb-cvs@lists.sourceforge.net] mailing list receives (in spite of its name) the subversion checkin messages. 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 [mailto:mhoenicka@users.sourceforge.net].

Part I. Overview

Table of Contents

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.

- RefDB can also serve as a data source for other applications. You can either use the SRU interface, or use the Perl and PHP libraries to integrate RefDB functionality directly into your own projects.

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.

1.3. General features

- RefDB uses a three-tier architecture (see Figure 1.1, "RefDB three-tier architecture"), 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
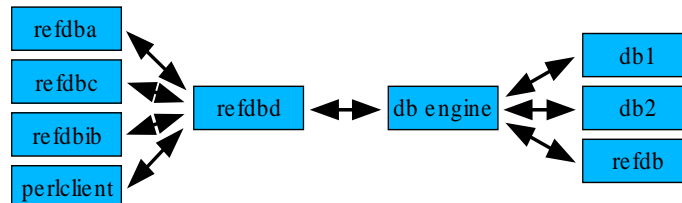
Optionally you may use an internal database engine instead of the external SQL server, resulting in a simpler two-tier setup.

- 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.

- By default, incoming data are converted to UTF-8 and stored as such (see Figure 1.2, "RefDB character encoding support"). Outgoing data can be converted to other character encodings if needed. However, RefDB can be set up in a very flexible manner to accommodate other character encoding requirements as long as they are supported by your operating system.

Figure 1.2. RefDB character encoding support
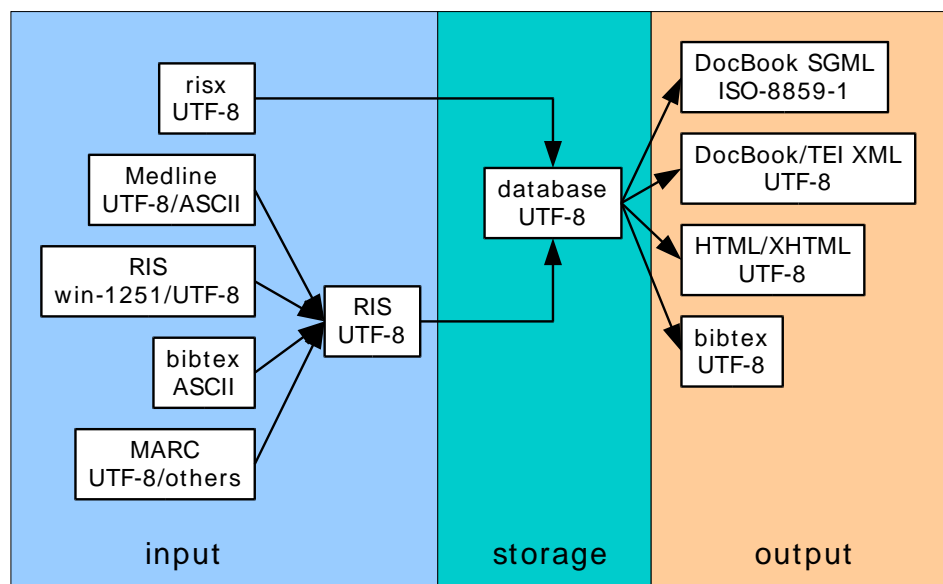
- The server can run as a daemon in a non-privileged account if security concerns require this. Besides, users can start it as a standalone application on demand. As just about everything is configurable, a thoughtful setup will allow to run several copies of the server on the same box in parallel.

1.4. Interfaces

- The RefDB command-line 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 command-line clients can be run in an interactive mode or in a batch mode. The latter is useful for scripting purpose or for use as backends of graphical user interfaces.

- A PHP interface allows to perform most RefDB operations through a web interface.

- RefDB also provides clients which allow to access RefDB databases through a SRU [http://www.loc.gov/standards/sru/] (Search and Retrieve via URL) interface. A simple standalone client for personal use and a CGI program for shared use through a web server are available.

- Separately available packages integrate RefDB 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 refdbd server without using the C clients. This allows the rapid development of custom programs that access RefDB databases.

1.5. Reference management features

- Reference data are imported from text files or from stdin and exported to text files or to stdout. The data model is fairly close to the *RIS* specification that most Windows-based reference managers understand. RefDB natively understands both the tagged RIS format used by other reference management software and an XML format according to the risx.dtd [http://refdb.sourceforge.net/risx/index.html] similar to RIS.

- Input filters can be used to convert references from various sources to the *RIS* format. Currently RefDB 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 RefDB 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://refdb.sourceforge.net/xnote/index.html].

- 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 (XML), MODS (XML), 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. RefDB 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.

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

- RefDB handles the L1 field of the RIS input files (denoting a local PDF or Postscript copy) in a very flexible way. You can specify a path to the file 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 personal reference lists. 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 addition to this default list, each user can create an unlimited number of additional lists.

  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.

1.6. Bibliography tool features

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

1.6.1. SGML and XML

- RefDB scans your input document for citations and generates a bibliography which contains the cited references. You can include this bibliography into your document and process it with RefDB's tools to obtain formatted citations and bibliographies.

- RefDB creates bibliographies for DocBook (SGML, DTD-based XML, and schema-based XML), TEI (DTD-based XML and schema-based XML), and bibtex.

- The implementation of the SGML/XML citations and bibliographies is "non-destructive", i.e. switching from one bibliographic output format to a different one is limited to re-creating the bibliography and transforming your document again.

  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" bibliography output to provide full control over the formatting. 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.

- Alternatively you can request "raw" bibliography output. In this case you can either rely on the default bibliography formatting provided by the stock DocBook or TEI stylesheet, or you resort to other bibliography formatting tools.

1.6.2. 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.

1.7. 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/utils/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 see fit).

- SQLite [http://www.sqlite.org] (version 2.8.x or 3.x.y): 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/utils/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 and the SRU servers are implemented as Perl scripts or Perl modules. These require a couple of external 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 and the refdb-perlsru packages on the refdb project page [http://sourceforge.net/projects/refdb].

   ## 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.

marc2ris

- MARC::Record (CPAN)

- MARC::Charset (CPAN)

- refdb-perlmod

med2ris

- XML::Parser (CPAN)

- Text::Iconv (CPAN)

- refdb-perlmod

en2ris

- Text::Iconv (CPAN)

- refdb-perlmod

refdbsru

- CQL (CPAN)

- CGI (CPAN)

- XML::Writer (CPAN)

- refdb-perlsru

- refdb-perlmod

refdb-sruserver

- CQL (CPAN)

- CGI (CPAN)

- XML::Writer (CPAN)

- HTTP::Server::Simple (CPAN)

- refdb-perlsru

- 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 refdbd. This allows to write custom clients that access the refdb databases and use the whole functionality implemented in refdbd. 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.

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 4.1.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 refdb_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. A web browser is of course also required to use the PHP interface, and it is also useful to interactively retrieve references through the SRU interface.

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

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. Both the standalone and the CGI version of the SRU services are also implemented in Perl. 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.

2.5. Related Software

Please visit the RefDB web page [http://refdb.sourceforge.net] for further information about the following tools which extend the capabilities of RefDB but are not included in the sources:

- ris-mode, an Emacs [http://www.gnu.org/software/emacs/] major mode for RIS datasets

- refdb-mode, an Emacs [http://www.gnu.org/software/emacs/] frontend for RefDB

- A package providing RefDB support for vim [http://www.vim.org], the improved vi editor

- refdb-perlmod, a package containing several Perl support modules for the Perl scripts shipped with RefDB

- RefDB::Client, a Perl [http://www.perl.org] module implementing an object-oriented access to RefDB for Perl programmers

- RefDB::SRU and RefDB::SRUserver, the Perl modules that implement the SRU access to RefDB

- A Ruby [http://www.ruby-lang.org] client library (partially implemented and currently unmaintained!)

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.

- 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.

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 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!

Part II. Administrator manual

Table of Contents

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 following sections of this chapter. The final sections deal with the setup of optional web-based frontends. 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.

Some of the above steps are simplified by the shell script refdb-init which you should have a thorough look at. Then 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. 0.9.9

- Major changes in both the reference databases and the main database require to upgrade both.

- Before re-importing your reference data, you may have to migrate the contents of some AV fields to L1

- Before re-importing custom citation styles, you may have to fix a few elements. The styles shipped with RefDB do not require these fixes, you can just add them again.

4.2.2. 0.9.8

- The implementation of personal reference lists has changed, most notably in that each user can have more than one of these lists. Although there was no need to alter the database schema, one of the

tables is now used to record the ownership of a reference instead of the membership of a reference in a personal list. The results may be a bit unexpected if you upgrade to 0.9.8 without migrating the data.

4.2.3. 0.9.7

- The main database needs to be re-created if you use MySQL as a database engine.

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 who 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, "RefDB on a standalone workstation"). 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


```
refdba      refdbd
refdbc      MySQL
refdbib
```
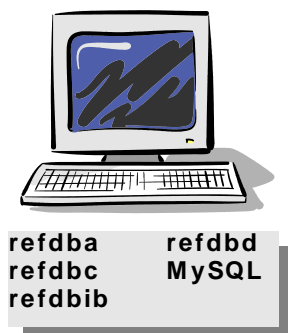
4.3.2.2. Installation in a network

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, "RefDB on a network"). 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
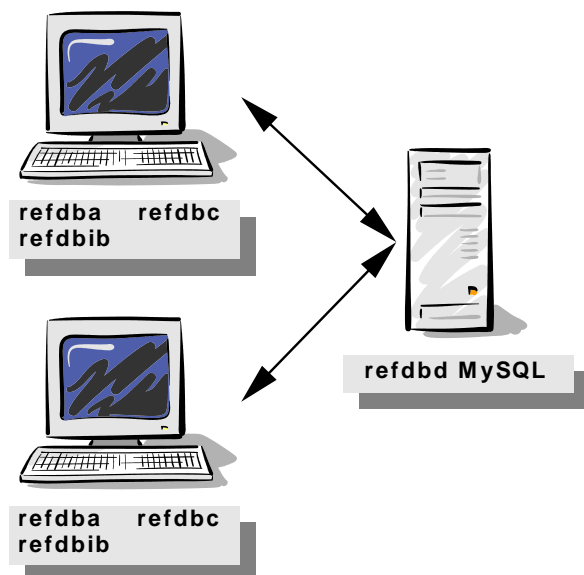
4.3.3. The mystery of the configuration files

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, and en2ris) 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).

bib2ris, marc2ris, and med2ris 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, marc2ris, and en2ris.

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 hardcoded 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.

- `sru` contains the XSLT and CSS stylesheets required to set up the SRU service

- `styles` contains some XML files containing bibliography styles.

- `xsl` contains XSLT stylesheets.

`/usr/local/var/lib/refdb/db`
: 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/utils/btOOL/] library.

  > **Note**
  >
  > Some operating systems/distributions use separate packages for the run-time libraries (ususally `.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-docbook-xsl-ns=PATH

Specify the full path to the root directory of the DocBook XSL-NS (for DocBook V5.0 and later) 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 for P4.

Note

As mentioned above for the DocBook stylesheets, use this option only if your catalogs cannot resolve the public identifiers properly. 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-tei-xsl-ns=PATH

Specify the full path to the root directory of the TEI XSL stylesheets for P5.

Note

As mentioned above for the DocBook stylesheets, use this option only if your catalogs cannot resolve the public identifiers properly. 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-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-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`. You may have to change this

name if you want to run two different versions of RefDB in parallel, or if you're not free to choose your database name. Although the main database name is configurable at runtime, you should use this option to initialize your `refdbrc` configuration file, as it allows the refdb-init script to use the proper database name.

--with-db-dir

Selects the directory which holds SQLite/SQLite3 databases.

--with-trang-jar

Specifies the full path to the jar file of the trang [http://www.thaiopensource.com/relaxng/trang.html] tool. You need this tool if you build RefDB from SVN sources, but not if you build from a tarball.

--disable-docs

RefDB ships with prebuilt docs. However, if you build a SVN version, or if the documentation is otherwise screwed up, you may have to build them. Building the docs from the sources requires a couple of extra tools. You can use this configure switch to build the rest of RefDB without having to install these tools.

--disable-clients

Use this switch if you want to build and install only the application server `refdbd`. This is mainly targeted at package builders.

--disable-server

Use this switch if you want to build and install only the clients but not the application server. This is mainly targeted at package builders.

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 SVN 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 SVN version is guaranteed to compile on the author's development platform (currently FreeBSD 6.1), 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. SVN 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
   svn co https://svn.sourceforge.net/svnroot/refdb/refdb/trunk refdb
   ```

   Other interesting modules to check out are `perlmod` (the Perl modules and the Perl client library) and `elisp` (the Emacs support files).

   If you are interested in a particular repository version, release, or branch, use commands along these lines:

   ```
   $~/build
   svn co --revision 324 https://svn.sourceforge.net/svnroot/refdb/refdb
   svn co https://svn.sourceforge.net/svnroot/refdb/refdb/tags/Release-0
   svn co https://svn.sourceforge.net/svnroot/refdb/refdb/branches/back-
   ```

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

   ```
   $~/build/refdb
   svn update
   ```

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

4. The SVN 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 and later 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.

5. The SVN version also lacks a few other things which are included in the source tarballs. In order to create the missing stuff, you'll need the following additional tools:

    • 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 instuctions) which generates the DTD documentation.

    • trang [http://www.thaiopensource.com/relaxng/trang.html]: a tool required to turn the RefDB DTDs into RelaxNG schemas.

    To work around problems, you may want to use the following **./configure** switches:

    • --with-trang-jar=PATH

    • --disable-docs

6. 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 SVN version up to date with a minimum number of keystrokes.

4.4.1.4. Installation of refdbd as a daemon

This is all it takes to run RefDB from the command line. If you want to start refdbd 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 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 refdbd 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 RefDB, 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 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
```

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
```

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-mysqlclient-lib=/usr/pkg/lib/mysql --with-btparse-lib=/us
```

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), PostgreSQL [http://www.postgresql.org] (version 7.1 or later), and SQLite/ SQLite3 [http://www.sqlite.org] (versions 2.8.x or later; 3.0.x 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.hml].

  SQLite does build on Cygwin, but all filename/path handling code uses the Windows conventions. You will have to use the Windows-style path when setting the database directory.

- 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, 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-docbook-xsl-ns=PATH

Specify the full path to the root directory of the DocBook XSL-NS (for DocBook V5.0 and later) 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 for P4.

Note

As mentioned above for the DocBook stylesheets, use this option only if your catalogs cannot resolve the public identifiers properly. 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-tei-xsl-ns=PATH

Specify the full path to the root directory of the TEI XSL stylesheets for P5.

Note

As mentioned above for the DocBook stylesheets, use this option only if your catalogs cannot resolve the public identifiers properly. 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-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-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`. You may have to change this name if you want to run two different versions of RefDB in parallel, or if you're not free to choose your database name. Although the main database name is configurable at runtime,

you should use this option to initialize your `refdbdrc` configuration file, as it allows the refdb-init script to use the proper database name.

--with-db-dir

Selects the directory which holds SQLite/SQLite3 databases

--with-trang-jar

Specifies the full path to the jar file of the trang [http://www.thaiopensource.com/relaxng/trang.html] tool. You need this tool if you build RefDB from SVN sources, but not if you build from a tarball.

--disable-docs

RefDB ships with prebuilt docs. However, if you build a SVN version, or if the documentation is otherwise screwed up, you may have to build them. Building the docs from the sources requires a couple of extra tools. You can use this configure switch to build the rest of RefDB without having to install these tools.

--disable-clients

Use this switch if you want to build and install only the application server `refdbd`. This is mainly targeted at package builders.

--disable-server

Use this switch if you want to build and install only the clients but not the application server. This is mainly targeted at package builders.

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=
--with-log-dir=/var/log --with-libdbi-dir=/uar/local/lib --with-docbook
--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 Windows' own tools: Either use the system control panel or use the 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. In most cases you will prefer to use a script provided with RefDB which performs all things required for a first-time installation. If you need more precise control, you'll also find instructions below how to set up RefDB manually.

4.8.1. Using the interactive setup script

The interactive script **refdb-init** must be run with root permission as it tries to fiddle with a couple of files that a regular user should not have write permission for. After starting the script, it will collect a few answers from you about your intended setup. It will also perform a couple of sanity checks. After this stage is completed, you will be asked to positively confirm whether or not your system should

be modified. Therefore it is safe to dry-run this script in order to decide whether you prefer the script over the manual installation.

The script creates the main database, the refdbd and refdba configuration files, loads the available styles, creates a reference database and a database user account. All that is left to do manually is to take care of the SGML catalog file.

4.8.2. Manual setup

4.8.2.1. 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.

The `refdbdrc` configuration file can be copied from one of the templates `refdbdrc.mysql.example`, `refdbdrc.pgsql.example`, `refdbdrc.sqlite.example`, or `refdbdrc.sqlite3.example` according to your choice of the database engine.

4.8.2.2. 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.

4.8.2.2.1. 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.

4.8.2.2.2. 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.

4.8.2.2.3. SQLite

The embedded database engine SQLite (versions 2.x) 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.

SQLite version 3.0 and higher uses UTF-8 as the default encoding which is just fine for the purposes of RefDB.

4.8.2.3. Creating the databases

The RefDB database contains common information that is shared by all reference databases. The following sections explain the database engine-specific steps. In all cases you have two options: The recommended way is to let **refdbd** handle the installation, but for special needs there is also a description how to set up the databases using the database engine clients.

4.8.2.3.1. MySQL

As mentioned above, the recommended way to create or update the main database is to run the following command from your root account:

```
~$
refdbd -a -u username -w password
```

Specify the username and password of your database administrator account.

If the default setup described above does not suit your needs, please use the following procedure instead to set up the main database.

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.

Instead of using **refdbd**, you can also use the hard way and create the main database using your database engine client:

- 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.mysql
```

For MySQL 4.1 and later, run this instead:

```
~$
mysql -u root refdb < /usr/local/share/refdb/sql/refdb.dump.mysql41
```

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.X.dump.mysql41|mys
```

Use the highest dump file number 'X' shipped with your current RefDB version. See the MySQL documentation [http://www.mysql.com] for further information about the table engines available with MySQL.

4.8.2.3.2. PostgreSQL

You must run the following command either from your root account, or from a special database administrator account (depending on your local PostgreSQL installation). Run this command to let **refdbd** create the main database:

```
~$
refdbd -a -u username -w password
```

Specify the username and password of your database administrator account.

If the default setup described above does not suit your needs, please use the following procedure instead to set up the main database.

> ## Note
>
> Your PostgreSQL administrator accound may not have the privileges to write to the PID file. In that case, add something like `-P  /var/tmp/refdbd.pid` to the command line mentioned above. 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 "pgsql" 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.

Instead of using **refdbd**, you can also create the main database using your database engine client:

- In a command line window, run the command:

```
~$
createdb -U pgsql -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 pgsql refdb < /usr/local/share/refdb/sql/refdb.X.dump.pgsql
```

Use the highest dump file number 'X' shipped with your current RefDB version. 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.

- Access control works through user groups. To be able to access the main database, create a RefDB user group like this:

```
~$ psql -U pgsql refdb -c "CREATE GROUP refdbuser"
~$ psql -U pgsql refdb -c "GRANT SELECT ON CITSTYLE, POSITIONS, REFSTYLE, SF
```

4.8.2.3.3. SQLite

Run the following command from your root account to let **refdbd** create the main database:

```
~$
refdbd -a
```

If the default setup described above does not suit your needs, please use the following procedure instead to set up the main database.

- The default database directory is `/usr/local/var/lib/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/var/lib/refdb/db
```

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

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

Use the highest dump file number 'X' shipped with your current RefDB version.

4.8.2.4. 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.

4.8.2.5. The shell scripts

The RefDB shell scripts and Perl scripts 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, RefDB 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 RefDB 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:

    error while loading shared libraries: libdbi.so.0: cannot open shared object file: no such file or directory
    > This is a common problem if you build your system from packages, but have to build some libraries (like libdbi, as shown here) from the sources. libdbi and most other autotools-based programs are installed into `/usr/local` by default. On some systems (like Debian, Ubuntu) the linker does not peek into `/usr/local/lib` for shared objects. To fix this, please add `/usr/local/lib` to `/etc/ld.so.conf` and run **ldconfig**(8), or configure the libraries to install into `/usr/lib` instead. See **./configure --help** for further instructions.

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 refdbd 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 refdba 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 PostgeSQL, 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:

```
#~
refdba
```

In the second case you have to use this instead:

```
#~
refdba -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 RefDB programs read their configuration files properly. To this end, first run the following refdba 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 refdbd. 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 refdbd variables, you're fine. This output may look like the following:

```
You are served by: RefDB 0.9.4
Client IP: 127.0.0.1
Connected via pgsql driver (dbd_pgsql 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.

A third possible outcome of this test is the message "main database too old or corrupt". Sometimes this is just what it says, e.g. if you upgraded to a newer version without re-creating the main database although the UPGRADING file told you to do so. However, the most likely reason is an incorrect setting in the refdbdrc config file. Please check that the dbserver variable matches the database engine that you created the main database for. The default is sqlite.

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.

4.10. SRU support

In addition to its native clients, RefDB also supports optional read access through a web-based search protocol called SRU [http://www.loc.gov/standards/sru/] (Search and Retrieve via URL). There are two options to enable SRU access. We'll first describe a CGI script which is the preferred way but requires a running web server on your system. The subsequent section describes a simpler standalone server for testing purposes and for single-user access.

4.10.1. Setting up SRU support as a CGI program

In order to provide SRU services at your site, you need to add a CGI [http://hoohoo.ncsa.uiuc.edu/cgi/] script to your web server. The following instructions assume that you use Apache [http://httpd.apache.org/] as your web server. However, the CGI script will work just fine with any other web server that provides CGI support.

4.10.1.1. Configure your web server

If your web server is already set up to run CGI scripts, you might be all set to run the SRU CGI script as well. If you're not sure, follow the instructions below and check whether the your configuration file already has the relevant entries.

The Apache web server uses a configuration file called `httpd.conf`. On many systems this file is located in `/usr/local/etc/apache`. Locate the section starting with "<IfModule mod_alias.c>" and make sure it declares a CGI directory:

```
ScriptAlias /cgi-bin/ "/usr/local/www/cgi-bin/"
```

You can choose a different directory, but the directory listed here must exist and be accessible from the account the web server runs in (often "www" or "nobody").

Now you have to allow the execution of CGI scripts in that directory:

```
<Directory "/usr/local/www/cgi-bin">
    AllowOverride None
    Options ExecCGI
    Order allow,deny
    Allow from all
</Directory>
```

Finally, you have to restart your web browser to let these changes take effect:

```
~# apachectl restart
```

4.10.1.2. Install the SRU CGI script

First you need to make sure the `/cgi-bin/` directory configured above exists and has the proper permissions. Local habits may vary, but you should see something like this:

```
~# ls -ld /usr/local/www/cgi-bin
drwxr-xr-x  2 root  wheel  512 Mar 13 14:56 cgi-bin
```

Note

In the Apache default installation `/usr/local/www/cgi-bin` is a symbolic link to an example CGI directory. Remove the symlink and create a real directory with the appropriate ownership and permissions instead.

Now copy the `scripts/refdbsru` CGI script into that directory and make it executable for all:

```
~# cp refdbsru /usr/local/www/cgi-bin
~# chmod a+x /usr/local/www/cgi-bin/refdbsru
```

All SRU replies are XML files. This is just fine if the requesting agent is a program that intends to further process the data, or to store them somewhere. If the requesting agent is a web browser with a human being in front of it, the plain XML output is a bit hard on the eyes though. It is recommended to provide XSLT stylesheets which can render the XML output in a human-readable HTML format (most current web browsers support XSLT these days). These stylesheets are used by default, unless a user provides the URL of a stylesheet of his own.

Unless you already have a designated folder for system-wide stylesheets, just create one, and make sure it is readable:

```
~# mkdir -p /usr/local/www/data/styles
~# chmod a+r /usr/local/www/data/styles
```

Now copy the example stylesheets shipped with RefDB (installed in `/usr/local/share/refdb/sru`) into that directory:

```
~# cp refdbsru.xsl mods.xsl risx.xsl refdbsru.css /usr/local/www/data/styles
```

Needless to say, you can tweak all these stylesheets to your heart's content.

Finally you should provide a configuration file called `refdbsrurc` which usually goes into `/usr/local/etc/refdb/`. You'll find a preconfigured example file in the same directory which you can copy and edit. Make sure the `xslurl` variable points to the `refdbsru.xsl` file that you've just installed. The value of this variable is an absolute or relative URL in the filesystem of the web server. If your setup is as described above, `/styles/refdbsru.xsl` will work just fine. It is recommended to set the variables starting with `zeerex_` to meaningful values. These values determine part of the output of the **explain** SRU command and are supposed to provide the users of your site with useful information about your SRU service.

4.10.1.3. Tune the SRU CGI script

From refdbd's point of view, the refdbsru CGI script is just another client. The script therefore requires a similar configuration file like the other clients, and it needs a database account to work with. For the configuration of the script, see the refdbsru reference.

As SRU provides only read access to your databases, it is recommended to create a separate database account for SRU accesses which all SRU users share. If you specify these values in the configuration files, the SRU users will not have to provide any login information. You may want to restrict that account to read access only using the appropriate switch of the adduser command.

4.10.1.4. Test the SRU CGI script

Open your favourite web browser and enter the following URL:

```
localhost/cgi-bin/refdbsru
```

You should now receive an XML document that describes the RefDB SRU service. If you receive a web server error instead, retrace your steps above and make sure you have restarted the web server before testing. If nothing else helps, please peruse the documentation of your web server. The appropriate document for Apache is the CGI HowTo [http://httpd.apache.org/docs/2.0/howto/cgi.html].

Finally, you should advertize the new service on your web site. Provide a link to "<hostname>/cgi-bin/refdbsru" along with some instructions how to use SRU, as explained in the user manual.

4.10.2. Setting up SRU support using a standalone web server

RefDB also provides a standalone web server to run the SRU service. All you need to do in terms of "installation" is to create or edit the SRU configuration file that the standalone server shares with the CGI application. See also the refdb-sruserver reference. Then start the SRU server like this:

```
~# [perl] refdb-sruserver
```

The server will run in the foreground until you kill it with **Ctrl**+**c**. Now point your web browser to:

```
localhost:8080
```

As you can see, the SRU server uses the non-standard port 8080 so it does not interfere with any regular web server. This also means that the standalone SRU server is usually not accessible from a

remote computer unless you configure your firewall appropriately. However, as long as you connect locally you should receive the output of the SRU explain command.

4.11. Install the PHP interface

If you want to provide interactive access to your RefDB databases through the web, consider installing the PHP web interface. This web frontend provides a convenient way to perform the most common user tasks, like running queries or adding datasets.

4.11.1. Prerequisites

Please check the following prerequisites on your computer before attempting to install the PHP interface

Web server

Any web server that is capable of running PHP scripts. If you don't run a web server anyway, Apache [http://httpd.apache.org] is always a good choice to start with. Both versions 1.3.x and 2.x will work just fine.

PHP interpreter

You need PHP5 [http://www.php.net], the older PHP4 won't do. When building PHP, make sure to configure the software appropriately so the php5 Apache module gets built. If you install a prebuilt package, you may have to select an extra package that contains the appropriate module for your web server.

With the exception of session support, the RefDB PHP interface relies solely on core features of PHP5. You may therefore have to install PHP5 session support separately. However, on some systems like FreeBSD, "core" does not mean "it's there". If the PHP5 core is split into several packages, you may also have to install DOM, XML, and XSL support separately.

4.11.2. Web server configuration

The following instructions assume that you use Apache as your web server. The configuration file is usually `/etc/apache/httpd.conf` or `/usr/local/etc/apache/httpd.conf`.

- Locate the section "Dynamic Shared Object (DSO) Support". Check if the php5 module is loaded anyway. If not, add a line like the following:

```
LoadModule php5_module    libexec/apache/libphp5.so
```

- The subsequent section in the config file needs to be updated too if you had to add the php5 module:

```
AddModule mod_php5.c
```

- Further down there should be a section containing Aliases. Create a new one within this section like this:

```
    Alias /refdb/ "/usr/local/share/refdb/www/"

    <Directory "/usr/local/share/refdb/www">
        AllowOverride None
        Order allow,deny
        Allow from all
```

```
        AddType application/x-httpd-php .php .phtml
    </Directory>
```

The paths point to the files that RefDB installs with all other shared files. If you intend to customize or modify the interface, you can copy the files to a directory in your home folder and change the directories in the config file accordingly.

- Finally you'll have to let the Apache server re-read it's configuration file:

```
~# apachectl restart
```

4.11.3. Test your PHP interface

To see the PHP interface in action, first make sure that the RefDB application server is running:

```
~# refdbctl start
```

Now start your web browser and point it to:

```
localhost/refdb/
```

Please note the trailing slash! Now you should be greeted by the login screen.

Chapter 5. refdbd administration

This chapter will explain the usage of refdba for the administration of refdbd. 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/LaTeX documents, the names of the databases should be treated as case-insensitive and should not contain colons (':').

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 -W 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 -W 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                              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
```

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 done with the deleteuser command. 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: deleteuser -d foo 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 refdba 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 refdba 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 all client-side tools 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 refdbd 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, refdb 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 refdb, e.g. the PostgreSQL client library, send log messages to stderr. This is not controlled by the log settings of refdbd. 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 refdbd 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

level	verbose	explanation
2	crit	the system is in a critical condition
3	err	there is an error condition
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 refdbd. The values are explained above.

process ID
: This is the process ID (as seen in the **ps ax** listing) of the refdb process that generated the message. In the case of refdbd 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 (but you can request the old behaviour using the $-x$ command line options since version 1.0). 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 refdb.

• 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 refdb three-tier design. Keep in mind that only refdbd communicates with the database server. Therefore only the host where refdbd 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 refdbd 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 PostreSQL do not allow remote connections for security reasons. You need to manually allow remote connections as described below.

• The refdba 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 administistators 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 refdbd application server. If you cannot restrict read access to

system tables for whatever reason, you should not enable refdbd remote administration (default is off) and use the **kill** command or the refdbctl script instead.

• Most of the refdba 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 refdbd 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 you 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.

Part III. User manual

Table of Contents

Chapter 6. Overview of the RefDB command-line 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 refdba, 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 are 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 refdba 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 refdbd is not installed to run as a daemon, start it now: **refdbd -s &** (see Starting refdbd for more options).

3. Start the refdbc 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 refdba and refdbc

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

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 refdba 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.

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 refdbib

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 refdbib. Keep in mind that the runbib script offers a simpler and more versatile way to run refdbib.

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 RefDB 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 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.

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 Reference Manager Manual [http://www.referencemanager.com/support/docs/ReferenceManager11.pdf] contains sort of a specification of the RIS data format. This 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:

```
❶
TY  - BOOK                    ❷
AU  - Hoenicka,M.             ❸
TI  - The RefDB manual
PY  - 2001
ER  -                         ❹
```

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 no built-in means to specify the character encoding of the data. Commercial applications usually expect the data to be supplied in a particular encoding (e.g. Reference Manager uses the Windows ANSI character set). 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 (UTF-8).

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 an 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 alphanumerical (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

Synonym: AB (this is not logical as it sounds like ABstract, but I'm sure there is a reason in RIS' crooked history). 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

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 a physical copy or offprint of 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 a call number.

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

L1 through L4

These fields hold the URLs of an offprint (e.g. a PDF or Postscript copy), of the fulltext (often available as HTML in addition to PDF), of a related resource, and of an image, respectively. Multiple entries of each type per reference are allowed. In contrast to the UR field above, these fields hold information which is stored for each user separately.

The L1 field is meant to store the location of local offprints. It is most common to use the "file://" protocol and specify a full path (starting with another slash on Unix systems) or a relative path. In the latter case the path is expanded with the pdfroot setting before it is displayed to the user. Using this feature requires some thought and is therefore explained in a separate section.

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, and the usage differs from one type to another. An important information among the data contained here is the digital object identifier [http://www.doi.org]. Please refer to the Reference Manager Manual [http://www.referencemanager.com/support/docs/ReferenceManager11.pdf] for the nasty details of these fields.

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 microenvirinment 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 /
Peitedes 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 helpers

If your editor of choice should be Emacs or XEmacs, the separately available refdb-elisp [http://
refdb.sourceforge.net/emacs.html] package will come in handy. Please see the installation instructions
contained in the source archive. The package consists of a major mode (ris-mode) to edit RIS datasets,
and a minor mode (refdb-mode) which provides a general front-end to RefDB. refdb-mode is too
powerful to be discussed here in a few lines, please see the separately available manual [http://
refdb.sourceforge.net/doc/elisp/index.html]. The ris major mode can do the following 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.

Note

> Once you have created an empty tag, you can use commands provided by the refdb-mode to enter values from existing entries in the database. This is convenient to avoid spelling errors in author names, or to select keywords which are already used by other references. These commands use minibuffer completion for added convenience.

- 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 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 displayed in the default foreground color (usually black on X displays), you have a problem as well.

7.3. Input filters

RefDB gets all reference data from text files (or from equivalent data read from stdin). 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.

refdb_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

A tool to convert Pubmed data in both the tagged and the XML format to RIS. You may use refdb-pubmed to retrieve Pubmed data in the first place.

en2ris

A tool to convert EndNote "RIS" exported data to RIS

bib2ris

A tool to convert BibTeX data to RIS

db2ris

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

marc2ris

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, which interconverts bibtex, COPAC, EndNote (Refer), EndNote (XML), ISI, PubMed, MODS, and RIS data. For example, use the following command to import MODS datasets into RefDB:

```
# xml2ris modsdata.xml|refdbc -C addref
```

7.4. Writing risx datasets

XML documents using the risx DTD [http://refdb.sourceforge.net/risx/index.html] 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/index.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 risx 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 risx documentation [http://refdb.sourceforge.net/risx/index.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 risx 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/xnote/index.html]. 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 refdbd 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, refid, 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 (type refid) or citation key (type reference). 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. Some special characters which are not allowed as part of an XML attribute value are stripped. Non-ASCII characters are converted to a reasonable ASCII equivalent, or they are dropped if no replacement is possible.

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 refdbc command line client. We will demonstrate the interactive use of refdbc, 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.

Example 1

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
```

Example 2

You feel the urge to import MODS data, an XML data format developed by the Library of Congress [http://www.loc.gov]. RefDB does not support this format natively or via converters at this time, so you'll have to resort to an external tool. Chris Putnam's bibutils [http://www.scripps.edu/~cdputnam/ software/bibutils/bibutils.html] package comes to the rescue, as it provides a set of command-line tools to interconvert a number of bibliography data formats. Both RIS and MODS are supported, so we have all it takes. The following command imports the MODS data into your default database in a single step, although you may first want to check the output of the converter before going ahead:

```
# xml2ris modsdata.xml|refdbc -C addref
```

8.1.2. risx datasets

If you prefer the XML data format instead, use the **addref** with the `-A 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 **xmllint**.

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.

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, or **xmllint**, which is shipped with libxml2.

```
~#
onsgmls -s /usr/local/share/refdb/declarations/xml.dcl foo.xml
~#
xmllint --noout --dtdvalid "http://refdb.sourceforge.net/risx/index.htm
```

No news is good news in both cases: 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 -A 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 (e.g. screen, HTML, or DocBook output) 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.

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 (but see the Emacs [http://refdb.sourceforge.net/emacs.html] and vim [http://refdb.sourceforge.net/vim.html] support files which allow you to add and update references from within your editor). The general 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. Managing personal reference lists

Personal reference lists mainly serve two purposes:

- If you share your reference database with other users, your default personal reference list records which datasets you've added. Coincidentally, the name of this list is identical to your database username.

- In addition to your default list, you can maintain as many additional lists as you see fit. Use these lists to select references for particular purposes, e.g. when preparing a presentation or if you need to keep track of papers that you still have to read.

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

8.6.1. Creating and deleting personal reference lists

Actually there are no commands to create or delete personal reference lists. These lists are created automatically as soon as you add references to them. 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 default 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. To add a reference to a list other than your default list, use the `-b listname` switch.

3. If you use the **updateref** command to modify the information in an existing reference, this reference will also appear in your default 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 **dumpref** 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. In both cases personal reference lists will be automatically removed if they are empty.

8.6.2. Accessing references in personal reference lists

You can limit your database search with the `-b listname` switch of the getref command to those references that you added to that particular personal reference list. If you want to access references in your default list, you have to pass your username as an argument to the `-b` switch.

8.6.3. Advanced use of personal reference lists

The pickref and dumpref commands mentioned above are actually only simplified interfaces to the extended notes feature. Each personal reference list corresponds to one extended note with these particular features:

• The note is initially almost empty. Only the title, the username and the key are set.

• The title and the key are initially identical and have the format <username>-<listname>. The username prefix is used to create a separate namespace for each user.

However, you can treat these notes just like any other note. In particular, you can retrieve the note with the getnote command, edit it by entering a description as the contents or a couple of keywords, and finally update it with the updatenote command. Just keep in mind that this information will be lost if you remove all references from this list, as this will cause the note representing the list to be purged automatically.

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

Scientific journals are very often cited by standardized abbreviations which help to keep the space required for bibliography listings at a minimum. However, at times you need the full name instead. If you did not add both synonyms when adding the references that use a journal, you can use the updatejo command to add or edit these synonyms.

8.9. Character encoding issues

The 7-bit ASCII character set originally employed by PCs 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 worthwile 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.

SQLite3
:   SQLite3 uses UTF-8 by default. UTF-16 is supported by the database engine, but not by the libdbi library which RefDB uses to access the engine.

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 (UTF-8).

Please note that the import filters med2ris, en2ris, and to a limited extent also marc2ris 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 refdbc 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 L1 field of a reference can hold the path to an electronic version of the document. 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 `file:///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 L1 field of the reference entry could then be set to `file://1999/2345.pdf` and `file://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. In that case it is most practical to just store the full URL in the L1 field, as in `http://some.machine/literature/Miller1999.pdf`.

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. Emacs [http://refdb.sourceforge.net/emacs.html] and vim [http://refdb.sourceforge.net/vim.html] can directly interact with RefDB using special modes or support files. Any other editor will do as well as long as it creates 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**.

Note

Some editors like Emacs [http://refdb.sourceforge.net/emacs.html] and vim [http://refdb.sourceforge.net/vim.html] support RefDB by means of add-on packages.

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/xnote/index.html]. 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.

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):

```
refdbc:
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.

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.

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/xnote/index.html] 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 manuscript or a thesis, 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.

RefDB's job is to provide two kinds of information:

• the bibliographic data

• styling information according to a bibliography and citation style

If the default rendering of citations and bibliographies in the DocBook or TEI stylesheets is appropriate for your purposes, you can get away with using RefDB as a source for raw bibliographies. However, if your output is supposed to match the requirements of a particular journal or publisher, you'll need the styling information as well. There are literally thousands of possible combinations for the formatting of authors, titles, journal names, page and date informations, and almost each of these possiblilities 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 documents with formatted citations and a formatted bibliography:

1. Load one or more bibliography styles into your RefDB database, using the addstyle command (this is usually done automatically during post-installation setup, see refdb-init).

2. The bibliography will eventually be available as a separate file (see below). SGML and XML documents have to include this file, either as an external entity or via xinclude. LaTeX or RTF documents need no special care at this point.

3. Insert citations into your document, preferrably using the short notation for SGML/XML documents. LaTeX documents use the regular bibtex commands, whereas RTF uses a plain-text citation format.

4. Run the appropriate commands to create the bibliography and to transform the document. For SGML, XML, and RTF documents this may be as easy as running **make pdf**, whereas LaTeX users have to run one extra command in addition to the usual bibtex procedure.

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 vs. citation key 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 bibliograpy 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 a validating XML editor like Emacs/nXML to write bibliography styles using the CiteStyle XML DTD. This ensures that you end up with a valid style that won't confuse RefDB. If you edit styles with a simpler editor, please run the resulting file through a validating parser like onsgmls or xmllint before adding it to your main database.

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, "Schematic representation of a `CITESTYLE` element"). 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
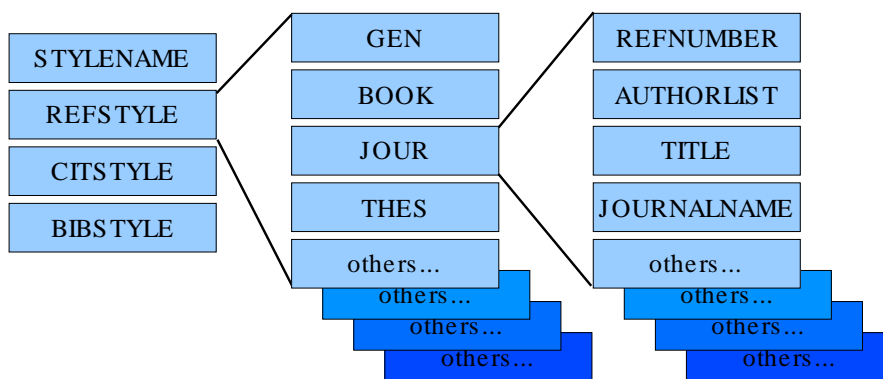
Each definition for a publication type in turn is basically an ordered list of the elements that make up the rendered bibliographic entry, like authorlists, publication dates, titles, and so on. You can arrange them

in any order you like. 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 styles also reflect the three-level representation of the bibliographic data themselves. For a discussion of this representation see the description of the risx format. When you write a bibliographic style, you have to make sure to pick the correct level, as indicated with the `role` attribute, for the author lists and the titles. E.g. a chapter entry would typically have `AUTHORLIST` and `TITLE` elements with the `role` attribute set to "PART" to display the chapter author and title, respectively, and additionally `AUTHORLIST` and `TITLE` elements with the `role` attribute set to "PUB" to render the editor and the title of the whole book, respectively.

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, and feel free to consult the styles shipped with RefDB for further guidance.

Tip

The RefDB project also provides a tool to create citation styles interactively. The **refdb-ms** Perl script is part of the RefDB sources.

10.3. Create SGML and XML bibliographies

Although SGML documents are usually processed with a DSSSL toolchain and XML documents with a XML toolchain, the procedures to generate documents with bibliographies are similar enough to treat them in a single section. If you use the high-level tools provided by RefDB, you won't even notice a difference.

RefDB can create two types of bibliographies, cooked and raw. Cooked bibliographies are already preformatted using the information of a particular bibliography and citation style. Documents using cooked bibliographies have to be processed using the appropriate stylesheet driver files provided by RefDB. Raw bibliographies (currently supported only for XML documents) contain no particular formatting and should therefore be processed using the regular DocBook or TEI stylesheets.

RefDB provides both high-level tools which attempt to hide the entire complexity of the bibliography business, and low-level tools which allow experienced users to integrate RefDB into their own toolchains. We'll first describe the high-level approach which should always be the first choice. The subsequent sections give all the details about the low-level tools which you normally don't even want to know about.

10.3.1. Keeping it simple with refdbnd

refdbnd provides the simplest approach to create, maintain, and transform documents with RefDB bibliographies. Once set up, all you'll have to do is to run something as simple as **make pdf**. The following subsections cover how to set up a refdbnd-managed project, how to cite, and how to process the document.

10.3.1.1. Setting up a project

refdbnd is an interactive script which creates a skeleton document and a custom-tailored `Makefile`. Start the script in a clean subdirectory by typing **refdbnd**. You'll be asked a couple of questions, each of which supplies sufficient background information for novice users. The script will then create three files (we'll assume that the basename that you provided was "foo"):

foo.short.[sgml|xml]

> The ".short" reminds you that you're supposed to use the short notation for citations in this file. This is simpler, usually more convenient, and should always be your first choice.

foo.[sgml|xml]

> This file is just a dummy in a fresh project. If you edit the file `foo.short.xml`, the file `foo.xml` will automatically be updated and fed to the subsequent processing steps next time you run **make**. Unless you know what you do, you don't want to touch this file.

Makefile

> This is a customized Makefile that contains all the information that you provided to refdbnd. Simply run commands like **make pdf** or **make html** to create printable or HTML output with formatted citations and bibliographies from your document.

10.3.1.2. Editing your document

The `foo.short[sgml|xml]` skeleton document contains the required markup to start a book or an article. You'll now want to open this file in your favourite text editor to write the contents and to add your citations. In DocBook SGML and XML documents, 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 values identifiy the bibliographic entries in your database. Use either numerical IDs as in the examples above, or alphanumeric citation keys as shown 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">2;5;9</seg>
```

Again, you can use citation keys instead of the numerical IDs 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.

10.3.1.3. Transforming your document

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. Not all FO processors offer RTF output though.

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.

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 clean && make pdf stylename="Eur.J.Pharmacol."
```

Note

> **make clean** removes intermediate files to let the change of the bibliography style take effect.

10.3.2. Bibliographies, the hard way

If the simple approach outlined above does not suit your needs, you can turn to the low-level bibliography tools provided by RefDB. Needless to say, you can always start with a refdbnd-created project and use the low-level tools whenever you run into any limitations. However, this section describes the manual creation and transformation of documents from the ground up.

10.3.2.1. Prepare the document

RefDB's bibliography output is a `bibliography` element that contains all required references. You can redirect the output into a file and include this file at the spot where your bibliography should appear. To achieve this you need two modifications in your document:

1. When using DTD-based documents (i.e. DocBook 4.x or TEI P4), 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://ww
     <!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.

When using schema-based documents (i.e. DocBook 5.x or TEI P5), there is no need to declare the bibliography at the beginning of the document.

2. Include the bibliography at the desired spot as an external entity for DTD-based documents:

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

Alternatively, use xinclude to include the bibliography in schema-based documents:

```
...
     <xi:include href="refdbtest.bib.xml"
     xmlns:xi="http://www.w3.org/2001/XInclude">
     <xi:fallback>refdbtest.bib.xml appears to be missing</xi:fallback>
     </xi:include>
```

Note

Some XSLT processors require a command-line switch or additional libraries to support xincludes. If your transformed document should lack the bibliographic listing, consult the documentation of your XSLT processor.

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.

10.3.2.2. 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 uses three 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.

ID notation for raw bibliographies
> If you want to process your document with the default Docbook or TEI stylesheets using a raw bibliography, you just use the citation key as te value of the `linkend` attribute of the `xref` or `biblioref` elements to refer to the appropriate reference entry in the bibliography. RefDB will generate the bibliography using the citation key as the `id` or `xml:id` (for DocBook 5.x and TEI P5) attribute.

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. Finally, we'll also show examples of the ID notation for raw bibliographies.

10.3.2.2.1. Short notation

The short notation has been described above as this is the notation which you use in refdbnd-maintained projects. The only thing you must not forget when not using refdbnd 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.

10.3.2.2.2. 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">        ❶
    <ptr targOrder="U" target="ID1-X" TEIform="ptr"/>        ❷
    </seg>
    <seg type="REFDBCITATION">
    <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)".
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 presenation 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">
    <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.

10.3.2.2.3. ID notation for raw bibliographies

This notation is similar to what you'd do in a document which does not use RefDB bibliographies at all, except that you have to declare which `citation` elements should be processed by RefDB (it may very well be that RefDB is supposed to process all of these elements, but in order to support cases where it shouldn't, there is a mechanism to allow just this). Just set the `role` attribute of the `citation` element to `REFDB` to include it in the list of citations that runbib extracts from your document, like this:

```
<citation role="REFDB"><xref linkend="Bellamy2002"/></citation>
```

10.3.2.3. 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.

10.3.2.3.1. 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.

To tell runbib that you want to create a raw instead of a cooked bibliography, use the `-r` command line switch. As there is no style information involved, you don't need the `-S` option in this case:

```
~$
runbib -d mybib -r -t db50x bar.xml
```

This will create a raw bibliography from the DocBook 5.0 document `bar.xml`, using the reference entries in the database `mybib`.

10.3.2.3.2. Do it the hardest possible way

The following steps trace back exactly what the runbib script does. 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 from SGML files (provide full paths as needed):

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

   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.

   XML files are processed using your favourite XSL processor. There are two different stylesheets available for raw and for cooked bibliographies. Both work all the same for DTD-based (DocBook 4.x, TEI P4) and schema-based (DocBook 5.x, TEI P5) documents:

```
#~
xsltproc --catalogs --xinclude /usr/local/share/refdb/xsl/citations.xsl  foo.xml > foo
#~
xsltproc --catalogs --xinclude /usr/local/share/refdb/xsl/citationsraw.xsl  foo.xml >
```

   In all cases 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 lines can be used to validate the document with (o)nsgmls or xmllint (change the paths as necessary):

```
~$
onsgmls -wxml -s /usr/lib/sgml/declaration/xml.dcl foo.id.xml
~$
xmllint --noout --nonet --dtdvalid file:///usr/local/share/refdb/dtd/citationlistx.dt
```

2. **Create the bibliography file**

```
~$
refdbib -d mybib -S "J.Irrep.Res." -t db31 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, refdbib 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.

If you want to generate a raw bibliography, use a command like this:

```
~$
refdbib -d mybib -r -t db50x foo.id.xml > foo.bib.xml
```

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.

refdbib 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 P4 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. TEI P5 bibliographies are exported directly by refdbd and do not require further processing.

10.3.2.4. Transform the document

Finally you can transform the document to create printable or HTML output. If you use cooked bibliographies 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.

Processing your document with a raw bibliography does not differ from processing any other DocBook or TEI document. However, you can still use the refdbxml script to avoid having to type the full command line of your XSL processor. Use something like this to process a Docbook 5.0 document with a raw bibliography:

```
~$
refdbxml -t pdf -s db5 bar.xml
```

The `-s` option tells the script to use the stock DocBook stylesheets for version 5. Other values are "db", "tei", and "tei5" for the DocBook stylesheet for version 4.x, the TEI P4 stylesheets, and the TEI P5 stylesheets, respectively

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.

10.3.3. How to use custom stylesheets

We have assumed in the previous instructions that the stock DocBook or TEI stylesheets suit your needs when processing your documents. However, if you need a particular formatting of the parts of your document which are not under the control of RefDB (things like fonts, colours, font sizes and so on), you'll have to create a driver file with your personal modifications, and somehow make sure this driver file is used whenever your document is processed. This section discusses the available mechanisms to use particular XSL driver files as a per-user or a per-document option. Currently no such mechanism is available for the DSSSL stylesheets.

A driver file is essentially a stylesheet which imports the stock stylesheets and adds a few modifications. XSL is designed such that any definition of a template in the driver file overrides the definition in the imported file. RefDB uses this mechanism extensively to provide the formatting of citations and bibliographic listings. Whenever you run refdbib (or runbib which calls the former), a driver file is created which imports a general RefDB driver file. This general driver file in turn includes the stock stylesheets (imports can be nested). If the stock stylesheets don't suit your needs, you'll have to provide two driver files:

- a driver file for the stock stylesheets which contains your modifications of those parts which are not under RefDB's control. Obviously, this driver file must import the stock stylesheets. You can put this driver file into any convenient subdirectory in your home directory. For further information about how to set up a driver file, please see Bob Stayton's DocBook XSL Guide [http://www.sagehill.net/docbookxsl/index.html] (the general approach is applicable to TEI and any other XSL stylesheets just as well)

- a modified general RefDB driver file which imports your driver file instead of the stock stylesheets. To this end, copy the relevant general RefDB driver file (there are driver files for fo, html, and

xhtml output, and they are available both for DocBook and for TEI) to a convenient subdirectory in your home directory and modify the import statements to suit your needs.

There are two options to have the modified general RefDB driver file used instead of the default ones:

- if you want to apply your modifications as a default to all documents which you process, consider adding the paths to your personal copy of the `runbibrc` configuration file.

- if you want to apply the modifications to a particular refdbnd-created project, just specify the paths of the modified general RefDB driver files when setting up the project. This way, each project can use a different set of modifications.

Figure 10.2, "Stylesheets involved in processing RefDB documents" visualizes how a document containing a RefDB bibliography is processed with a focus on the stylesheets involved. The example shows the transformation of a DocBook XML document to fo (which might then be processed to e.g. PDF). However, the same principles are applicable to other output formats and other document types. The left hand side shows the default processing using the stylesheets installed by RefDB. refdbib (which may be invoked by runbib, or by running a refdbnd-created `Makefile`) creates an intermediate stylesheet containing the style-specific information. This stylesheet is converted to output-type-specific driver files for fo, html, and sometimes xhtml output. The figure shows only the fo driver file to keep it simple. This driver file imports the general RefDB fo driver file, which in turn imports the appropriate official DocBook fo stylesheet. The right hand side shows how to process the same document with a DocBook driver file which may alter the general formatting of the document (page size, borders, fonts and the like). As you can see, you have to provide both the DocBook driver file and an equivalent of the general RefDB fo driver file, which has to import your driver file instead of the stock DocBook stylesheet.

Note

You can of course add all your general modifications to `myrefdbdriver.xsl` and have that import the stock DocBook stylesheet. However, doing it in two steps as shown will allow you to use `mydocbookdriver.xsl` for non-RefDB projects as well.

Figure 10.2. Stylesheets involved in processing RefDB documents
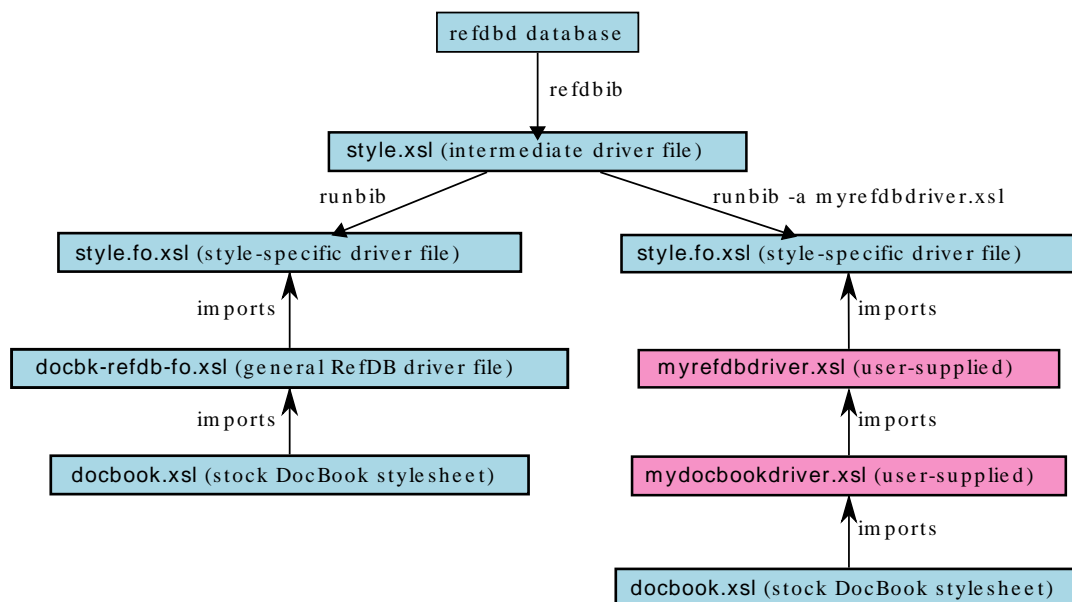
10.4. Create LaTeX/BibTeX bibliographies

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, refdbib 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 citation key of the reference and tell the processing application which database to use. Or you specify the database name with each citation. In this case, you can pull the references from different databases in the same document. The two versions look like this:

   ```
   \cite{Miller1999}
    \cite{litibp:Myers2001}
   ```

   The first version cites the reference with the citation key "Miller1999" in the database passed to the processing application as an argument. The second form cites the reference with the citation key "Myers2001" in the database "litibp". Please note that, in contrast to SGML/XML citations, the database part is separated by a colon from the citation key.

   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:

```
#~
sort foo.aux | uniq | refdbib -d mybib -S name -t bibtex > foo.bib
```

   ## Note

   The `.aux` file should be preprocessed through sort and uniq as shown here to avoid duplicate entries in your bibliography.

   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 refdbib 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
```

10.5. Create RTF bibliographies

Rich Text Format (RTF) is a plain-text format understood by most word processors. While RefDB does not integrate into the menu bar of M$ Word or OpenOffice, it still allows you to add bibliographies to word processor documents saved as RTF files. This is a one-way process which leaves your original document untouched. You can edit the compound document and save it to native word processor formats like `.doc` or `.odt`, but you'll have to start over with the RTF document as soon as you add, change, or remove citations. Therefore the following sequence (which will be familiar to the SGML/XML and LaTeX folks) is recommended when using RefDB to create bibliographies for word processor documents:

1. **Author your document**

Write the contents of your document until you're *really done*. Use the citation format described further down. It does not matter which file format you use at this stage, as long as you can export it to RTF in the end.

2. **Save your document in RTF format**

Now save your document as a RTF file. There may be an extra menu entry called Export but usually you can just select the file format in the Save as... dialog.

3. **Create and insert the bibliography**

   The exact procedure will be described below. The runbib command creates a bibliography file, and the refdbrtf tool combines the source document and the bibliography file to a new compound RTF document. Note that neither your word processor document nor the RTF copy are altered by this procedure.

4. **Import and finalize your document**

   You can now open the compound RTF document using your word processor and add final touches (like adding images which you don't need during the authoring step). You can save the document in a word processor format, print it, or export it as a PDF file.

As with SGML and XML documents, RTF documents can be maintained in a simple fashion using a refdbnd-created Makefile. The more complex way of running the involved tools manually essentially parallels the way used for SGML and XML documents and will not be elaborated here.

10.5.1. Create a RTF document for use with RefDB

In contrast to SGML and XML documents, there is nothing special about RTF documents that you can use with RefDB. You should still run refdbnd and select "rtf" as a document type. This will generate a suitable Makefile and a skeleton RTF document. You can open that with your favourite word processor, or copy an existing RTF document and save it under the same name.

10.5.2. Create citations in word processor documents

RefDB recognizes a simple plain-text citation format in RTF documents. Citations are enclosed in square brackets. Inside a citation, each reference is again enclosed in square brackets. References are identified by their citation keys, followed by one of "-X", "-A", and "-Y" to denote regular citations, author-only citations, or year-only citations, respectively (the RefDB low-level tools handle the first vs. subsequent occurrence issue silently for you, therefore there is no need for "-S" and "-Q" although these are valid too). The following text snippet shows some citations:

```
...was shown[[Miller1999-X]]. This was confirmed by other groups
as well [[Doe2000-X][Jones2000-X]]. However, [[Nerd2002-A]] challenged this
view in a recent work[[Nerd2002-Y]]...
```

10.5.3. Process RTF documents

As word processor documents lack the separation between content and formatting, there is usually no transformation required to read or print a document. However, as the processing requirements from RefDB's point of view are not any different between RTF and XML documents, RefDB still has to transform your RTF document - to yet another RTF document. While doing so, it massages your in-text citations into appropriately formatted links to the bibliographic entries, and appends a formatted bibliographic listing. All you need to do is to run **make**. This will create the output file `foo.refdb.rtf` from your project file `foo.rtf`.

10.6. Using custom stylesheets to process documents with bibliographies

The RefDB stylesheets handle the oddities of formatting your bibliographies. The stock DocBook or TEI stylesheets handle the remainder of your documents. What if the formatting of the latter is not to your liking? To this end, the recommended way is to use stylesheet driver files which override those parameters, or entire sections, of the stylesheets which you need modified. Now, processing

documents containing RefDB bibliographies with regular DocBook or TEI driver files won't do you any good as you'll lose the journal-specific citation and bibliography support. However, you can easily design driver files which accomplish the task at hand.

refdbnd-created Makefiles contain hooks to use custom driver files; if you prefer to run the transformations manually, these Makefiles will also tell you how. The relevant Makefile section looks like this:

```
# options to use customized RefDB driver files, if any
fodriveropt = -a custom-fo.xsl
htmldriveropt = -b custom-html.xsl
xhtmldriveropt = -c custom-xhtml.xsl
```

As shown here, you can pass options to runbib which specify custom driver files. You can keep those in the same directory as the document itself, or specify a relative or full path. Your custom driver files *must* include the appropriate RefDB stylesheets, not the stock DocBook or TEI stylesheets, like this (shown for the fo driver file):

```
<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  version="1.0">

  <xsl:import href="/usr/local/share/refdb/xsl/docbk-refdb-xsl/docbk-fo/docbk-r

  <!-- your customizations here -->

</xsl:stylesheet>
```

Chapter 11. RefDB SRU interface

11.1. What SRU is all about

SRU [http://www.loc.gov/standards/sru/] (Search and Retrieve via URL) is a standard search protocol developed by the Library of Congress [http://www.loc.gov] to allow web-based access to libraries. SRU uses a fairly simple query language called CQL [http://www.loc.gov/standards/cql/index.html] (Contextual Query Language). Any program that can send a HTTP GET request to a remote site and receive the resulting XML document is basically suited as a SRU client. This is especially true for your web browser and for command-line utilities like wget [http://www.gnu.org/software/wget/], but a dedicated client like YAZ [http://www.indexdata.dk/yaz] may be more convenient for all but the most simple queries. Also, some software packages read bibliographic data from databases via SRU (e.g. citeproc [http://xbiblio.sourceforge.net/citeproc/]).

RefDB implements SRU server capabilities through a CGI script that you can optionally install in your web server. This will allow anyone with access to your web server to search and retrieve bibliographic data in your RefDB databases. Alternatively, a standalone web server for single-user access is available too. This chapter explains how to use SRU access. See the installation chapter for the instructions how to set up SRU support in your RefDB installation.

11.2. SRU Operations

Note

This section assumes that you run the SRU service using the CGI application. If you use the standalone server instead, please adapt the URLs by replacing "http://mybox.com/cgi-bin/" with "http://localhost:8080/".

SRU defines three operations, all of which return XML documents:

explain
  describes the available facilities in terms of record schemas, available indexes and so on. Sort of a cheat sheet. The original specification is here [http://www.loc.gov/standards/sru/explain/].

searchRetrieve
  performs a database query and retrieves the matching datasets. The original specification is here [http://www.loc.gov/standards/sru/sru-spec.html].

scan
  retrieves a list of matching search terms for later use in a searchRetrieve operation. The original specification is here [http://www.loc.gov/standards/sru/scan/index.html].

You are encouraged to peruse the linked specifications above to learn the general principles. The following sections build on this knowledge and describe the RefDB SRU interface with a focus on its peculiarities and limitations. We'll assume that your web server is set up to run the `refdbsru` CGI script using the following URL: http://mybox.com/cgi-bin/refdbsru/.

11.2.1. The explain operation

The explain operation is the simplest of all and a good start to introduce the syntax of the SRU interface. RefDB fully supports the explain operation. Any of the following URLs typed into your browser will run it:

• http://mybox.com/cgi-bin/refdbsru/

• http://mybox.com/cgi-bin/refdbsru/?

- http://mybox.com/cgi-bin/refdbsru/?operation=explain&version=1.1

The URL part following the question mark ("?") in the third example is the search-part which consists of "parameter=value" pairs glued together with ampersands ("&"). Both parameters shown here are mandatory for all SRU operations as we'll see shortly.

The query will return a XML document describing the capabilities of the RefDB SRU interface.

11.2.2. The searchRetrieve operation

The searchRetrieve operation is the one used to actually get hold of the reference data you're looking for. Your query is sent in the `query` parameter which is mandatory for this operation. A few examples:

- http://mybox.com/cgi-bin/refdbsru/?
  operation=searchRetrieve&version=1.1&recordSchema=mods&query=bib.name%3d
  %22Miller,Henry J.%22

- http://mybox.com/cgi-bin/refdbsru/?
  operation=searchRetrieve&version=1.1&recordSchema=risx&query=dc.subject%3d%22circular
  dichroism%22+or+dc.subject%3d%22NMR%22

The first example requests the bibliographic data in MODS format. The query proper reads 'bib.name="Miller, Henry J."' and translates to a search for all references where a person with that name is listed as an author, editor, or series editor. The second example requests the data in risx format and searches all references with the keywords "circular dichroism" or "NMR".

Both examples make use of percent encoding to make the URL string conform to the specs. This is further discussed below.

11.2.2.1. The query parameter

The query parameter describes the criteria of your database query and is a string using the Common Query Language.

11.2.2.1.1. Conformance

The RefDB SRU support conforms to CQL Level 2 [http://www.loc.gov/standards/sru/cql/index.html#conformance]. The following general restrictions apply:

- RefDB does not support persistent result sets. Therefore, the `resultSetTTL` request parameter is meaningless, and it is not possible to reference a result set in a subsequent query.

- RefDB does not support XPath expressions to modify the results. Therefore the `recordXPath` request parameter is not honored. You can of course apply any XPath expressions on the client side using an appropriate processor.

- Sorting is currently not supported, and the `sortKeys` parameter is not applicable. Data will always be sorted by ID

- The `recordPacking` parameter is not supported. Records are always returned as XML.

- RefDB does not support relation modifiers and boolean modifiers in CQL queries.

- `prox` is not supported as a boolean operator.

- The relation `encloses` is not supported

- The support for regular expressions ("masking" in CQL) depends on the database backend. Most notably, anchoring is not supported by SQLite and SQLite3.

11.2.2.1.2. Defaults

If a query or a query part does not specify an index and a relation, RefDB looks for the term in the author, keyword, and title indexes:

http://mybox.com/cgi-bin/refdbsru/?operation=searchRetrieve&version=1.1&query=cat

This query will try to find references that contain the string "cat" in either the title, a keyword, or an author name.

11.2.2.1.3. Context sets

RefDB supports the context sets Dublin Core [http://www.loc.gov/standards/sru/cql/dc-context-set.html] (dc) and the not yet officially released CQL Bibliographic Searching [http://www.loc.gov/standards/sru/cql-bibliographic-searching.html] (bib). The following table lists the relationship of the indexes defined in these context sets with the RefDB fields.

Note

RefDB of course implicitly also supports the cql context set.

Table 11.1. Context sets

dc index	bib index	RefDB field	search/scan?	description
title	title	TX	y/n	item titles
	seriesTitle	T3	y/n	series title
	titleAbbrev	JA	y/y	journal title, abbreviated
creator, contributor	name, namePersonal, nameCorporate	AX	y/y	authors and editors
subject, coverage	subject	KW	y/y	keywords
date	dateIssued	PY	y/n	publication date
	volume	VL	y/n	periodical volume
	issue	IS	y/n	periodical issue
	startPage	SP	y/n	start page
	endPage	EP	y/n	end page
publisher		PB	y/n	publisher

11.2.2.1.4. Encoding

As you may have noticed, it is necessary to percent-encode [http://rfc.net/rfc3986.html#s2.1.] a few special characters in the parameter *values*. E.g. the equal sign ("=") assigns the values to the parameters and does not have to be encoded. However, equal signs within the CQL query string (which is the value of the `query` parameter) must be percent-encoded. If you use a dedicated client to run your queries, you should not have to care about these conversions. If you use a web-browser or a similar device, you may find the following conversion table useful:

Table 11.2. Percent-encoding special characters

replace	with	replace	with
:	%3a	/	%2f
?	%3f	#	%23

replace	with	replace	with
[	%5B	]	%5D
@	%40	!	%21
$	%24	&	%26
'	%27	(	%28
)	%29	*	%2a
+	%2b	,	%2c
;	%3b	=	%3d
%	%25	"	%22

11.2.2.2. Schemas

RefDB can return the datasets using two different XML schemas which you can request with the *recordSchema* parameter:

MODS

MODS [http://www.loc.gov/standards/mods/] is a schema for bibliographic data in library applications. Use 'mods' as the parameter value. The returned datasets will use 'mods' as the namespace prefix. MODS is the default if you do not specify a schema.

risx

This is RefDB's default XML input and output format. Use 'risx' as the parameter value to request risx. The datasets will use 'risx' as the namespace prefix.

11.2.2.3. Databases

SRU assumes that the base URL of the SRU service (the one you enter to get an explain response) corresponds to one database. Instead of using several copies of the CGI script to service more than one database, refdbsru allows to specify the name of a database in the additional path information of the URL. Compare the following (pseudo-)URLs:

```
http://myserver.com/cgi-bin/refdbsru/?<query>
http://myserver.com/cgi-bin/refdbsru/foo?<query>
```

The first URL will use the default database. The second URL will use the database "foo" instead. The database name goes between the slash that follows the CGI script name and the question mark that opens the query string.

11.2.3. The scan operation

The purpose of the scan operation is to provide a matching list of query terms, along with the number of references each term would retrieve. This is similar to browsing through a stack of library cards with subjects or author names on them. The RefDB SRU service allows to scan the following database fields:

- keywords (bib.subject)

  http://mybox.com/cgi-bin/refdbsru/?operation=scan&version=1.1&scanClause=bib.subject%3d%22dichroism%22

- author names (bib.name)

  http://mybox.com/cgi-bin/refdbsru/?operation=scan&version=1.1&scanClause=bib.name%3d%22Henry J.%22

- journal abbreviations (bib.titleAbbrev)

Part IV. Reference manual

Table of Contents

Chapter 12. The application server

refdbd is the RefDB application server, i.e. a program that runs in the background and handles the requests from the RefDB clients. This is the program that directly interacts with the database server and does most of the serious work.

refdbd can be configured at startup with the configuration file `refdbdrc` or by passing command-line options. The latter override the corresponding settings in the configuration files. While refdbd is running, most of the parameters can be changed remotely with refdba (see the refdba command **confserv**), but to make changes permanent you'll have to edit the configuration file or the script line that starts refdbd.

Warning

In the current implementation, the remote administration via refdba 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 **refdbctl reload** instead to cause refdbd 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 refdbd 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 refdbd with **Ctrl-C**. The `-s` command line switch lets refdbd 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 refdbd as a NT service is to use a service installer as described in the Installation chapter. In this case you should run refdbd 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 refdbd from Cygwin inetd. In that case you'd run refdbd as a daemon again, but this is currently beyond the scope of this manual.

You can run refdbd 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.

This chapter 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 section. If you need further control, you can use the command line of refdbd directly as explained in the second section. The third section describes the script which is usually run automatically by your system if you install refdbd as a daemon. The final sections describe the two applications which provide SRU access to RefDB databases.

Name

refdbctl — refdbd startup script

Synopsis

refdbctl *start|stop|restart|reload*

Description

refdbctl is a wrapper script to be used as a manual control script for the refdbd(1) daemon. It is also used by the refdb(8) startup script. The script takes care of avoiding multiple copies of the daemon and allows you to stop the daemon without knowing its process ID.

Options

start
> Starts the refdbd(1) daemon unless it is already running

stop
> Stops the refdbd(1) daemon

restart
> Restarts the refdbd(1) daemon if it is already running. The existing process is stopped, and a new process is started.

reload
> Asks the refdbd(1) daemon to reload its configuration file. Use this command to let changes to your configuration file take effect without stopping the process.

Files

`/var/run/refdbd.pid`

The run file containing the process ID of the running refdbd process.

See also

RefDB (7), *refdbd* (1).

RefDB manual (local copy) PREFIX/share/doc/refdb-<version>/refdb-manual/index.html

RefDB manual (web) <http://refdb.sourceforge.net/manual/index.html>

RefDB on the web <http://refdb.sourceforge.net/>

Author

refdbctl was written by Markus Hoenicka <markus@mhoenicka.de>.

Name

refdbd — the application server of RefDB

Synopsis

```
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][-x][-y confdir][-Y libdbidir]

refdbd {[-a] | [-c]} [-b dbs-port] [-D dbserver] [-e log-destination] [-E
encoding][-h][-i IP-address][-l log-level][-L log-file][-p port][-P PID-
file][-q][-u username][-v][-V][-w password][-x][-y confdir][-Y libdbidir]
```

Description

refdbd is the application server of RefDB(7). refdbd contains most of the application logic of RefDB and interacts with the database engine. refdbd must run somewhere in your network to do anything useful with the RefDB clients. refdbd usually runs as a daemon and responds to client requests, but it can be started as a regular process for debugging purposes. To start refdbd as a server, use the first command synopsis shown above.

It is recommended to use a wrapper script to start and stop refdbd. If you want to start and stop refdbd manually, use refdbctl(1). If you want to run refdbd as a daemon, use refdb(8). Edit the configuration file (see below) to permanently configure refdbd.

In addition to being run as a server, refdbd can also be invoked to check, install, or upgrade the main database. Refer to the second command synopsis shown above.

Options

-a

Runs refdbd to install or upgrade the main database. refdbd will exit after performing the maintenance tasks. In order to perform the maintenance tasks you have to provide a database administrator username and password using the -u and -w options, respectively, if your database engine uses access control. Some database configurations (e.g. PostgreSQL on Debian) further require you to run the administrative tasks from a special privileged user account (often called pgsql or postgres). If you use one of the file-based engines (SQLite or SQLite3), you must run the tool from an account which has write permissions in the database folder. refdbd first checks whether a main database already exists. If not, it will attempt to install it. Otherwise, it will upgrade the database to the current version if required. refdbd will print an error message to stderr if the maintenance job fails, and exit with a non-zero exit code.

-b dbs-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.

-c

Runs refdbd to check the main database version and the database engine connection. refdbd will exit after performing the maintenance tasks. In order to perform the maintenance tasks you have to provide a database administrator username and password using the -u and -w options, respectively, if your database engine uses access control. If you use one of the file-based engines (SQLite or SQLite3), you must run the tool from an account which has read permissions for the main database file. refdbd will print an error message to stderr if the check fails, and exit with a non-zero exit code.

-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 *log-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. Specify the IANA name of the encoding. 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 *log-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.

-L *log-file*
> This switch specifies a custom log file (full path please). This will only be used if the -e switch is set accordingly.

-p *port*
> Set the port on which refdbd listens for incoming connections. The default is 9734.

-P *PID-file*
> 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.

-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 *note-share-mode*
> 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 *name*
> Set the username of the database administrator account.

-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.

-w *password*
> The password of the database administrator account. You can pass an asterisk to let refdbd ask for a password interactively. This keeps your password from showing up in the process list. Keep in mind that you have to protect the asterisk on the command line by surrounding it with single quotes.

-x
> Assume incoming passwords are unencrypted.

-y *confdir*
> 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 *libdbidir*
> Specify the directory where the libdbi drivers are Note: By default, libdbi (the database abstraction library used by refdbd) 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.

Configuration

Table 12.1. 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	sqlite	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/var/lib/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.
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).
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 *a lot*). -1 means nothing will be logged.
pidfile	/var/log/refdb.pid	The file refdbd writes its process ID to. If you start refdbd from the command line as a regular user,

Variable	Default	Comment
		you should specify a file that you have write access to (you may not be allowed to create `/var/log/refdbd.pid` or write to this file as a regular user).
port	9734	The port on which refdbd 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 refdbd 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 refdba. 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 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.
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

Variable	Default	Comment
		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.
no_decrypt	f	If set to 't', incoming passwords are assumed to be unencrypted. The default is to expect encrypted passwords.

Files

`PREFIX/etc/refdb/refdbdrc`

The global configuration file of refdbd.

See also

RefDB (7), *refdb* (8), *refdbctl* (1), *refdba* (1), *refdbc* (1).

RefDB manual (local copy) PREFIX/share/doc/refdb-<version>/refdb-manual/index.html

RefDB manual (web) <http://refdb.sourceforge.net/manual/index.html>

RefDB on the web <http://refdb.sourceforge.net/>

Author

refdbd was written by Markus Hoenicka <markus@mhoenicka.de>.

Name

refdb, refdb.sh — refdbd startup script

Synopsis

refdb *start|stop|restart|force-reload*

Description

refdb is a wrapper script for refdbd(1) to be used as a rc(8) or init(8) control script. The script is called refdb.sh on BSD-style systems. See the documentation of your system how to integrate refdb into the start process to run refdbd(1) automatically as a daemon.

Options

start
   Starts the refdbd(1) daemon

stop
   Stops the refdbd(1) daemon

restart
   Restarts the refdbd(1) daemon

force-reload
   Causes the refdbd(1) daemon to read its configuration file

See also

RefDB (7), *refdbd* (1) *refdbctl* (1).

RefDB manual (local copy) PREFIX/share/doc/refdb-<version>/refdb-manual/index.html

RefDB manual (web) <http://refdb.sourceforge.net/manual/index.html>

RefDB on the web <http://refdb.sourceforge.net/>

Author

refdb was written by Markus Hoenicka <markus@mhoenicka.de>.

Name

refdbsru — CGI script providing SRU access to RefDB databases

Synopsis

```
refdbsru
```

Description

refdbsru is a CGI script which turns your web server into a proxy RefDB server providing SRU (Search and Retrieve via URL) access. Anyone with a web browser or some other tool capable of sending HTTP requests can query your RefDB databases. This script is not intended to be run by users. It is invoked by your web server.

The output of the script is a XML document as described in the SRU standard.

Configuration

refdbsru evaluates the file `refdbsrurc` to initialize itself.

Note

The parameters starting with "zeerex_" do not actually influence how the script operates. They provide some of the site-specific information which appears in the SRU **explain** output.

Table 12.2. refdbsrurc

Variable	Default	Comment
logfile	/var/log/refdbsru.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.
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 refdbs run on the same machine.
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.
timeout	180	The timeout in seconds. After this time has elapsed, a

Variable	Default	Comment
		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.
passwd		The password which is used for authentication with the database server.
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.
cssurl	(none)	In spite of the parameter name, this may be the URL either of a Cascading Style Sheet (CSS) file, or of a XSLT file. This file, if specified, is used to format the SRU output in the web browser.
dbserver	(none)	This is the name of the database engine that refdbd uses (currently either mysql, pgsql, sqlite, or sqlite3). refdbsru can obtain this value at runtime, but it has to send one additional query per searchRetrieve or scan operation. If you provide the value here instead, response times will be faster.

Variable	Default	Comment
zeerex_host	www.change.me	The hostname of the computer that runs your web server and refdbsru
zeerex_port	80	The port that your web server listens to.
zeerex_database	cgi-bin/refdbsru	The path section of the URL which provides SRU access to your database, without the leading slash. zeerex_host, a slash, and this option combined make up the URL which a user has to type into his web browser to access your SRU service.
zeerex_databaseInfo_title	Reference Database	A public name of your database.
zeerex_databaseInfo_description	A reference database	A brief description of your database.
zeerex_databaseInfo_author	Your Name	The name of the person who is in charge of running the database.
zeerex_databaseInfo_contact	your@email	How to contact the database maintainer. Usually the email address.

Environment

As a CGI script, refdbsru evaluates a variety of environment variables that the web server provides. These are:

REQUEST_METHOD
> Either GET or POST

QUERY_STRING
> The query string proper in a GET request.

PATH_INFO
> This value, if present, is used as the name of a database (other than the preconfigured default database).

Files

PREFIX/etc/refdb/refdbsrurc
> The global configuration file of refdbsru.

See also

RefDB (7)

refdb-sruserver (1)

RefDB manual (local copy) PREFIX/share/doc/refdb-<version>/refdb-manual/index.html

RefDB manual (web) <http://refdb.sourceforge.net/manual/index.html>

RefDB on the web <http://refdb.sourceforge.net/>

SRU (Search and Retrieve via URL) <http://www.loc.gov/standards/sru/>

Author

refdbsru was written by Markus Hoenicka <markus@mhoenicka.de>.

Name

refdb-sruserver — Simple SRU server for RefDB databases

Synopsis

```
refdb-sruserver
```

Description

refdb-sruserver is a simple standalone web server which povides a SRU service for RefDB databases. It is intended solely for local single-user access. For anything more ambitious, please see the CGI version of the SRU service, refdbsru(1).

Before starting the SRU service, make sure to set up the refdbsrurc configuration file which it shares with the CGI variant. The RefDB application server refdbd(1) must also be up and running. Now start the script which will run in the foreground until you kill it with Ctrl-c. The SRU service is available at the URL http://localhost:8080. Pointing your web browser to this address should return the output of the SRU explain command in a nicely formatted page. To run real queries, append the SRU query strings to the above URL.

Configuration

refdb-sruserver evaluates the file `refdbsrurc` to initialize itself. It shares this configuration file with refdbsru(1), the CGI variant of the SRU service.

Note

The parameters starting with "zeerex_" do not actually influence how the script operates. They provide some of the site-specific information which appears in the SRU **explain** output.

Table 12.3. refdb-sruserverrc

Variable	Default	Comment
logfile	/var/log/refdb-sruserver.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.
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 refdbs run on the same machine.
port	9734	The port on which refdbd listens. Change this for all clients and the server if this value interferes

Variable	Default	Comment
		with another program using this port.
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.
passwd		The password which is used for authentication with the database server.
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.
cssurl	(none)	In spite of the parameter name, this may be the URL either of a Cascading Style Sheet (CSS) file, or of a XSLT file. This file, if specified, is used to format the SRU output in the web browser.
dbserver	(none)	This is the name of the database engine that refdbd uses (currently either mysql, pgsql, sqlite, or sqlite3). refdb-sruserver can obtain this value at runtime, but it has to send one additional query per searchRetrieve or scan

Variable	Default	Comment
		operation. If you provide the value here instead, response times will be faster.
zeerex_host	www.change.me	The hostname of the computer that runs your web server and refdb-sruserver
zeerex_port	80	The port that your web server listens to.
zeerex_database	cgi-bin/refdb-sruserver	The path section of the URL which provides SRU access to your database, without the leading slash. zeerex_host, a slash, and this option combined make up the URL which a user has to type into his web browser to access your SRU service.
zeerex_databaseInfo_title	Reference Database	A public name of your database.
zeerex_databaseInfo_description	A reference database	A brief description of your database.
zeerex_databaseInfo_author	Your Name	The name of the person who is in charge of running the database.
zeerex_databaseInfo_contact	your@email	How to contact the database maintainer. Usually the email address.

Files

PREFIX/etc/refdb/refdbsrurc
   The global configuration file of refdb-sruserver.

See also

RefDB (7)

refdbsru (1)

RefDB manual (local copy) PREFIX/share/doc/refdb-<version>/refdb-manual/index.html

RefDB manual (web) <http://refdb.sourceforge.net/manual/index.html>

RefDB on the web <http://refdb.sourceforge.net/>

SRU (Search and Retrieve via URL) <http://www.loc.gov/standards/sru/>

Author

refdb-sruserver was written by Markus Hoenicka <markus@mhoenicka.de>.

Chapter 13. Administration tools

The following tasks can be performed with the tools described in this chapter:

- Perform the initial system setup

- 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

- Create and restore database backups

The first section explains the usage of the command-line client refdba in full detail. The following sections introduce two scripts which simplify the task of backing up and restoring the system and reference databases of your installation.

Name

refdba — the administration client of RefDB

Synopsis

Interactive mode:

```
refdba [-c pager-command] [-e log-destination] [-f stdin] [-h ] [-i IP-
address] [-l log-level] [-L log-file] [-p port] [-q ] [-T time] [-u name] [-v ] [-
V ] [-w password] [-x ] [-y confdir]
Non-Interactive mode:
```

Batch mode:

```
refdba -C command [-c pager-command] [-e log-destination] [-f stdin] [-i IP-
address] [-l log-level] [-L log-file] [-p port] [-q ] [-T time] [-u name] [-w
password] [-x ] [-y confdir]
```

Description

refdba is a command-line client providing the commands to administer RefDB(7) databases, users, and styles. refdba can be started in an interactive mode, providing a command prompt. Type *?* or *help* to see a list of available commands. Alternatively you can start refdba in non-interactive mode. refdba will execute the requested command and return. In this mode refdba will accept input on stdin for a variety of commands, allowing Unix piping.

Options

-c *pager-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.

-C *command*

The command to be run in non-interactive mode. You can supply all options and parameters that the command accepts on the refdba command line.

-e *log-destination*

log-destination 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.

-f *stdin*

Read data from stdin. refdbc usually knows when it should read from stdin. However, a few commands use data supplied in the command line but also allow to read from a file. Use this option to force refdbc to read from stdin *in addition* to values supplied on the command line.

-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(1). Instead of the IP address you can also specify the hostname as long as it can be properly resolved by your system.

-l *log-level*

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 *log-file*

Specify the full path to a log file that will receive the log messages. Typically this would be /var/log/refdba.

-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. Useful for debugging config files.

-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.

-x

Send passwords unencrypted.

-y *confdir*

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.

Diagnostics

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.

Configuration

refdba evaluates the refdbarc configuration file at startup to initialize itself.

Table 13.1. refdbarc

Variable	Default	Comment
logfile	/var/log/refdba.log	The full path of a custom log file. This is used only if logdest is set appropriately. If you start refdba 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/refdb.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 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 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

Variable	Default	Comment
		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.
no_encrypt	f	If set to 't', passwords are transmitted unencrypted. The default is to encrypt passwords.

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.

addstyle

Synopsis

addstyle [-c *command*] [-h] [[-o *filename*] | [-O *filename*]] {*style-file*...}

Description

Adds one or more bibliography style specifications from the input file(s).

Options

-c *command*
Specifies 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.

-h
Displays the online help about the **addstyle** command.

-o *filename*
Write the output to *filename* instead of to stdout.

-O *filename*
Append the output to *filename* instead of writing it to stdout.

style-file
All other arguments are interpreted as the names of files containing style specifications.

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.

adduser

Synopsis

```
adduser {-d database}[-h]{-H host-IP}[-R][-W password]{[-f file]|[username
...]}
```

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 -W 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.

-d database
    Specifies the reference database for which the access rights should apply.

-f file
    Reads a whitespace-separated list of usernames from `file`.

-h
    Displays the online help about the **adduser** command.

-H hostname
    `hostname` specifies 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.

-R
    Use this option to grant read-only access for the user. By default, users are granted read/write access. Users with read-only access can basically only retrieve references and notes.

-W password
    Set 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.

username
    All other arguments are interpreted as 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.

Examples

```
refdba:
adduser -d db1 -N newpass jim
```

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".

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.

addword

Synopsis

```
addword [-h] {[-f file] | [word...]...}
```

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.

Options

-f file
: Read a whitespace-separated list of journal title words from file.

`-h`
> Displays the online help about the **addword** command.

`word`
> All other arguments are interpreted as 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 which case you provide these words in.

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.

confserv

Synopsis

```
confserv {command} [value]
```

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.

Example

```
refdba: confserv loglevel 7
```

This will set the log level to 7. This temporary change will only be effective until refdbd is restarted.

createdb

Synopsis

```
createdb [-E encoding] [-h] {dbname...}
```

Description

Creates a new database with the name *dbname*. Several databases may be specified in a single call of this command.

Options

-E *encoding*
>   Select 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.

-h
>   Displays the online help about the **createdb** command.

name
>   The name of the reference database. The name must not contain a colon (':') or a dash ('-') due to the citation formats in documents using RefDB. The allowed characters may be further restricted by the database engine you use. The database name should also be considered case-insensitive, i.e. don't try to create a database "mybase" if you already have one called "MYBASE". Also, avoid using names which are SQL reserved words as this is doomed to fail. Unfortunately, this includes the all too convenient name "references". Try "refs" or "biblio" instead.

>   ### Tip
>   Prepend a constant string like "rd" to all refdb database names. This speeds up retrieving refdb databases with the **listdb** command if your database engine manages additional, non-RefDB databases. Use a simple regular expression like "rd%" to restrict your search to RefDB databases.

Example

```
refdba:
createdb db1 -E UTF-8 db2
```

This will create the databases db1 and db2 with the character encoding UTF-8.

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.

deletedb

Synopsis

```
deletedb [-h] {dbname...}
```

Description

Deletes the database with the name *dbname*. Several databases may be specified in a single call of this command.

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.

Options

-h
Displays a brief usage message and returns to the prompt.

dbname
The name of the database to be deleted.

Example

```
refdba:
deletedb db1 db2
```

This will delete the databases db1 and db2.

deletestyle

Synopsis

```
deletestyle [-h] {unix-regexp}
```

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.

Options

-h
Displays a brief usage message and returns to the prompt.

unix-regexp
The remaining arguments are interpreted as a regular expression which specifies the style or styles to be deleted.

Example

```
refdba:
deletestyle J\..*
```

This will delete all bibliography styles that start with "J.".

deleteuser

Synopsis

```
deleteuser {-d database} [-h] {-H host-IP} {-R} {[-f file]|[username ...]}
```

Description

Revokes access rights to a refdb database from the given users.

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.

Options

-d `database`
Specify the name of the database.

-f `filename`
Read the usernames from `filename`

-h
Displays the online help about the **deleteuser** command.

-H `hostname`
Specify the hostname or IP address for which to modify the access rights. 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 useother database engines.

-R
Revokes read-only access.

`username`
All other arguments are interpreted as 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.

Examples

```
refdba:
deleteuser -d -H % db1 jim
```

This will revoke the access to the database `db1` for the user jim for all but local connections.

deleteword

Synopsis

```
deleteword [-h] {[-f file] | [word ...] ...}
```

Description

This command performs the reverse operation of addword. The specified reserved words will be removed from the list.

Options

`-f`
> Read a whitespace-separated list of words from `file`.

`-h`
> Displays the online help about the **addword** command.

`word`
> All other arguments are interpreted as 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.

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.

getstyle

Synopsis

```
getstyle [-c] [-h] [[-o] | [-O]] { style ...}
```

Description

Retrieves one or more bibliography style specifications from the database and formats them as an XML file.

Options

`-c command`
> 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.

-h
: Displays the online help about the **getstyle** command.

-o
: Write the output to a file instead of to stdout.

-O
: Append the output to a file instead of writing it to stdout

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.

style
: All other arguments are interpreted as the names of bibliography styles.

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.

help

Synopsis

```
help

?
```

Description

Displays a brief summary of the available commands.

Example

```
refdba:
help
```

listdb

Synopsis

```
listdb [-h] [database-regexp]
```

Description

Lists all available databases if no argument is specified. If *database-regexp* is specified, only the databases matching this expression will be listed.

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.

Options

-h
Displays a help message explaining the **listdb** command.

database-regexp
A valid SQL regular expression which limits the output to matching database names.

Example

```
refdba:
listdb db%
```

This will list all databases with names that start with the string "db".

liststyle

Synopsis

```
liststyle [-h] [ style-regexp ]
```

Description

Lists all available bibliography styles that match *style-regexp*. If no argument is given, *all* available styles will be listed. This may or may not be what you want.

Options

-h
Displays a help message explaining the **listdb** command.

style-regexp
A valid Unix regular expression which limits the output to matching style names.

Note

Some database engines, like SQLite, do not support Unix-style regular expressions. Use SQL regular expressions instead.

Example

```
refdba:
liststyle ^J.*
```

This will list all bibliography styles that start with a capital "J".

listuser

Synopsis

```
listuser {-d database} [-h] [name-regexp]
```

Description

Lists all available users of the specified database that match `name-regexp`. If no argument is given, *all* available users will be listed. This may or may not be what you want.

Options

`-d database`
Specify the database name.

`-h`
Displays a help message explaining the **listdb** command.

`name-regexp`
A valid Unix regular expression which limits the output to matching database user names.

Note

Some database engines, like SQLite, do not support Unix-style regular expressions. Use SQL regular expressions instead.

Example

```
refdba:
listuser -d refs ^mo.*
```

This will list all users of the database "refs" whose names start with "mo".

listword

Synopsis

```
listword [-h] {word-regexp}
```

Description

Lists all available reserved journal words that match `unix-regexp`. If no argument is given, *all* available words will be listed. This may or may not be what you want.

Note

Keep in mind that the journal words are uppercased internally. You should write your `unix-regexp` using all caps accordingly.

Options

`-h`
Displays a help message explaining the **listdb** command.

`word-regexp`
A valid Unix regular expression which limits the output to matching journal title words.

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.

Example

```
refdba:
listword ^BIO.*
```

This will list all reserved journal words that start with "BIO".

scankw

Synopsis

```
scankw {-d database} [-h]
```

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.

Options

`-d database`
Specify the database name.

`-h`
Displays a help message explaining the **listdb** command.

set

Synopsis

```
set [-h] [varname] [varvalue]
```

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:"".

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.

Options

`-h`
Displays a help message explaining the **listdb** command.

`varname`
The name of the variable whose value should be displayed or set.

`varvalue`
The new value of the variable to be set.

Example

```
refdba:
set timeout 90
```

This command will set the timeout to 90 seconds for the current session.

verbose

Synopsis

```
verbose [-h]
```

Description

Toggles the verbose mode on or off. If the verbose mode is on, the error messages and warnings may be some more comprehensible.

Options

`-h`
Displays a help message explaining the **listdb** command.

Example

```
refdba:
verbose
```

Depending on the previous setting, this will toggle the verbose mode on or off.

viewstat

Synopsis

```
viewstat [-h]
```

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**.

Options

-h
Displays a help message explaining the **listdb** command.

Example

```
refdba:
viewstat
```

This will print some connection statistics and informations on the screen.

Files

PREFIX/etc/refdb/refdbarc
The global configuration file of refdba.

$HOME/.refdbarc
The user configuration file of refdba.

See also

RefDB (7), *refdbd* (1), *refdb-backup* (1), *refdb-restore* (1), *refdbc* (1).

RefDB manual (local copy) PREFIX/share/doc/refdb-<version>/refdb-manual/index.html

RefDB manual (web) <http://refdb.sourceforge.net/manual/index.html>

RefDB on the web <http://refdb.sourceforge.net/>

Author

refdba was written by Markus Hoenicka <markus@mhoenicka.de>.

Name

refdb-backup — RefDB tool: back up RefDB references, notes and styles

Synopsis

`refdb-backup [-h] [-u user] [-w pwd] [-f format] [-d dir] [-s ]`

Description

refdb-backup is a tool for backing up RefDB(7) references, notes and styles. A gzipped tarfile of the form 'refdb_backup_YYYYMMDD_HHMM.tar.gz' is created.

While users can restore data from the backup archive manually, it is expressly designed to restored by the companion tool *refdb-restore*.

Options

`-d dir`
> directory in which to place backup file (default: current directory)

`-f format`
> references storage format ('ris' (default)|'risx')

`-h`
> print help and exit

`-s`
> silent, no screen feedback

`-u user`
> username for RefDB clients (refdba, refdbc) not required if clients are configured for automatic access

`-w pwd`
> password for RefDB clients (refdba, refdbc) not required if clients are configured for automatic access

See also

RefDB (7), *refdb-restore* (1), *refdba* (1), *refdbc* (1).

RefDB manual (local copy) PREFIX/share/doc/refdb-<version>/refdb-manual/index.html

RefDB manual (web) <http://refdb.sourceforge.net/manual/index.html>

RefDB on the web <http://refdb.sourceforge.net/>

Author

refdb-backup was written by David Nebauer <david@nebauer.org>

This manual page was written by David Nebauer <david@nebauer.org> for the Debian project (but may be used by others).

Name

refdb-restore — RefDB tool: restore RefDB references, notes and styles

Synopsis

`refdb-restore` [-h] [-u `user`] [-w `pwd`] `backup-archive`

Description

refdb-restore is a tool for restoring RefDB(7) references, notes and styles. The backup archive must have been created by the companion tool *refdb-backup*.

Before performing the restore all reference databases must be removed (archived) from /var/lib/refdb/db/. The system database in that directory ('refdb') must be recreated from the sql dump (see *Manual*).

Options

-h
> print help and exit

-u `user`
> username for RefDB clients (refdba, refdbc) not required if clients are configured for automatic access

-w `pwd`
> password for RefDB clients (refdba, refdbc) not required if clients are configured for automatic access

See also

RefDB (7), *refdb-backup* (1), *refdba* (1), *refdbc* (1).

RefDB manual (local copy) PREFIX/share/doc/refdb-<version>/refdb-manual/index.html

RefDB manual (web) <http://refdb.sourceforge.net/manual/index.html>

RefDB on the web <http://refdb.sourceforge.net/>

Author

refdb-restore was written by David Nebauer <david@nebauer.org>

This manual page was written by David Nebauer <david@nebauer.org> for the Debian project (but may be used by others).

Name

refdb-init — RefDB setup script

Synopsis

```
refdb-init
```

Description

refdb-init is an interactive shell script that guides the system administrator through the initial setup of RefDB(7). refdb-init asks all necessary questions and performs all necessary checks before any changes are committed. Your confirmation is required before your system will be touched. As many of the operations performed by this script require root permission, you have to run this script from a privileged account.

While refdb-init does check for an existing installation of RefDB and offers to back up existing configuration files, it is not yet designed to upgrade an installation appropriately. If used to upgrade an existing installation, the administrator is responsible to back up and restore data that may be overwritten during the setup.

refdb-init performs the following tasks (you can interactively skip some of the tasks):

- Creates the main database refdb for the selected database engine. If a copy of the main database already exists, it asks for permission to delete this database before proceeding.

- Creates a configuration file for refdbd(1), the RefDB application server. If the file already exists, it will keep a copy of this version.

- Creates a configuration file for refdba(1), the RefDB administrative client, in the current login account. If the file already exists, it will keep a copy of this version.

- Starts refdbd, the RefDB application server

- Loads the bibliography and citation styles shipped with RefDB

- Creates a reference database

- Creates a database user with permissions to work with the reference database

Files

PREFIX/etc/refdb/refdbdrc
    The configuration file for refdbd

/root/.refdba
    The configuration file for refdba for the root account

See also

RefDB (7), *refdbd* (1).

RefDB manual (local copy) PREFIX/share/doc/refdb-<version>/refdb-manual/index.html

RefDB manual (web) <http://refdb.sourceforge.net/manual/index.html>

RefDB on the web <http://refdb.sourceforge.net/>

Author

refdb-init was written by Markus Hoenicka <markus@mhoenicka.de>.

Name

refdb-bug — RefDB tool: writes text file containing summary of RefDB's configuration

Synopsis

```
refdb-bug
```

Description

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 as *Reference Manager* or *EndNote* do for word processor documents.

This script creates a text file (./refdb-bug.txt) containing useful information about RefDB's configuration. This file can be attached to bug reports to aid in debugging.

See also

RefDB manual (local copy) PREFIX/share/doc/refdb-<version>/refdb-manual/index.html

RefDB manual (web) <http://refdb.sourceforge.net/manual/index.html>

Author

RefDB was written by Markus Hoenicka <mhoenicka@users.sourceforge.net>.

This manual page was written by David Nebauer <david@nebauer.org> for the Debian project (but may be used by others).

Chapter 14. Tools for reference and notes management

The following tasks can be performed with the tools described in this chapter:

- Manually add, update, and delete reference entries

- Search for and display reference entries

- Search for and display authors and keywords

- Include existing references into personal reference list

- Convert reference data from various formats to RIS

We'll first introduce the command-line client **refdbc**, RefDB's own multi-purpose reference and notes management client, along with the reference data converters that you can use along with it. The following sections explain the input and output data formats as well as the query language used by RefDB.

14.1. Tools

The first subsection explains the usage of the command-line client refdbc.

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. These utilities create tagged RIS data from the input data.

Note

Please visit Chris Putnam's Bibutils [http://www.scripps.edu/~cdputnam/software/bibutils/] project for another set of bibliographic data converters. These tools allow to convert a few additional formats like ISI and MODS to RIS which can then be imported by RefDB.

Since version 0.9.3, RefDB 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 RefDB 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 RefDB will assume the data are encoded as UTF-8.

Name

refdbc — the reference and note management client of RefDB

Synopsis

Interactive mode:

```
refdbc [-c pager-command][-d database][-e log-destination][-f stdin][-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][-x
][-y confdir]
```

Batch mode:

```
refdbc -C command [-c pager-command][-d database][-e log-destination][-f
stdin][-F fields][-g deffile][-G CSS-file][-i IP-address][-l log-level][-L
log-file][-p port][-q ][-R pdfroot][-T time][-u name][-v ][-V ][-w password]
[-x ][-y confdir]
```

Description

refdbc is a command-line client providing the commands to manage references and notes with
RefDB(7). refdbc can be started in an interactive mode, providing a command prompt. Type *?* or *help*
to see a list of available commands. Alternatively you can start refdbc in non-interactive mode. refdbc
will execute the requested command and return. In this mode refdbc will accept input on stdin for a
variety of commands, allowing Unix piping.

Options

-c *pager-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.

-C *command*

The command to be run in non-interactive mode. You can supply all options and parameters that
the command accepts on the refdba command line.

-d *database*

The name of the default database. You can change the database anytime during an interactive
session.

-e *log-destination*

log-destination 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.

-f *stdin*

Read data from stdin. refdbc usually knows when it should read from stdin. However, a few
commands use data supplied in the command line but also allow to read from a file. Use this
option to force refdbc to read from stdin *in addition* to values supplied on the command line.

-F *fields*

Specify the default fields that are to be displayed in a *getref* query.

`-g` *deffile*

> This option can be used to add some default fields to all RIS 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.

`-G` *CSS-file*

> 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*).

`-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(1). Instead of the IP address you can also specify the hostname as long as it can be properly resolved by your system.

`-l` *log-level*

> 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` *log-file*

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

`-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. Useful for debugging configuration files.

`-R` *pdfroot*

> Specify the root path of your collection of electronic offprints.

`-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.

`-x`

> Send passwords unencrypted.

`-y` *confdir*

> 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.

Diagnostics

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.

Configuration

refdbc evaluates the `refdbcrc` configuration file at startup to initialize itself.

Table 14.1. refdbcrc

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.
defaultris	(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.

Variable	Default	Comment
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
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 refdbc 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.
serverip	127.0.0.1	The IP address or hostname of the machine where refdbd runs. Use the default (localhost)

Variable	Default	Comment
		address if the clients and refdbs 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.
no_encrypt	f	If set to 't', passwords are transmitted unencrypted. The default is to encrypt passwords.

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.

addlink

Synopsis

```
addlink [-d database] [-h] [[-c command] | [-o outfile] | [-O outfile]] {note-
specifier} {link-target...}
```

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).

Options

-c command
> Pipe the output through command.

-d database
> Specify the database.

-h
>   Display a help message explaining the command.

-o *filename*
>   Write the output to `filename` instead of to stdout.

-O *filename*
>   Append the output to `filename` instead of sending it to stdout.

note-specifier
>   Specify one note by either its :NID: or its :NCK: value.

link-target
>   Specify one or more link targets by means of their :ID:, :CK:, :AU:, :KW:, :JF:, :JO:, :J1:, or :J2: values.

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".

addnote

Synopsis

```
addnote [-d database] [-E encoding] [-h] [[-c command] | [-o outfile] | [-O outfile]] [ file ...]
```

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.

Options

-c *command*
>   Pipe the output through *command*.

-d *database*
>   Specify the database.

-E *encoding*
>   Select the character encoding for the input data if it is different from the default UTF-8.

-h
>   Display a help message explaining the command.

-o *filename*
>   Write the output to `filename` instead of to stdout.

-O *filename*
>   Append the output to `filename` instead of sending it to stdout.

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.

addref

Synopsis

addref [-d *database*] [-E *encoding*] [-h] [[-c *command*] | [-o *outfile*] | [-O *outfile*]]
[-t *type*] [-U *username*] [*file* ...]

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.

Options

-c *command*
  Pipe the output through *command*.

-d *database*
  Specify the database.

-E *encoding*
  Select the character encoding for the input data if it is different from the default setting. 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 not needed and ignored.

-h
  Display a help message explaining the command.

-o *filename*
  Write the output to `filename` instead of to stdout.

-O *filename*
  Append the output to `filename` instead of sending it to stdout.

-t *type*
  Select the input data type. Possible values are "ris" (default) and "risx". Other data types have to be converted to one of these types before adding them to the database.

-U *username*
  Provide a different username than that of the current user, so e.g. some technician or administrative staff can add references in behalf of a researcher.

file
  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.

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.

checkref

Synopsis

`checkref [-A output-type] [-d database] [-E encoding] [-h] [[-c command]|[-o outfile]|[-O outfile]] [-r field-list] [-s field-list] [-t input-type] [-U username] [file ...]`

Description

Adds the references in `file` to temporary tables in the current database. The command is similar to the addref command, except that it does not add the references permanently to your database. Instead, the import is "simulated" in temporary tables, and the resulting datasets are analyzed in terms of similarities to existing permanent entries. If a reference is similar to an existing one in terms of the location (periodical, volume, issue, startpage), of the titles, or of the citekey, you may want to check these references manually as they are probably duplicates. If an abbreviated periodical name is reported to match an existing full name, you may want to add both names to the new reference to make refdb aware that it is dealing with the same journal. If an author name using abbreviated first or middle names is reported to be a possible duplicate of an existing author or vice versa, you may want to check if these authors are indeed identical, and change the abbreviated one to the full version. In addition, keywords are checked for similar existing keywords (often there are singular and plural forms of the same keyword). You should prefer to use existing keywords if possible to make your database more consistent and easier to search.

Options

`-A outtype`
    Select the output type of the report. Currently supported values are "scrn" for a terse screen output, and "xhtml" for a voluptuous xhtml report, bells and whistles included.

`-c command`
    Pipe the output through `command`.

`-d database`
    Specify the database.

`-E encoding`
    Select the character encoding for the input data if it is different from the default setting. 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 not needed and ignored.

`-G cssfile`
    Select the CSS stylesheet that is to be used for the xhtml output.

`-h`
    Display a help message explaining the command.

`-o filename`
    Write the output to `filename` instead of to stdout.

`-O filename`
    Append the output to `filename` instead of sending it to stdout.

-r *fieldlist*

    Select fields to check. If this option is not used, all available checks are performed. This may result in a more comprehensive report than you want. You can instead check for particular fields, or a subset of the available fields. *fieldlist* is a concatenation of the two-letter (pseudo) field codes: TX (all titles), PY (pubdate, volume, issue, pages), AX (all authors), JO (all journal names), CK, and KW.

-s *fieldlist*

    Select additional fields to display with the default xhtml output. *fieldlist* is a concatenation of the two-letter field codes of those fields which are not printed by default: N1, N2, NX, AB, AD, PB, CY, RP, SN, LX, U1-U5, and M1-M3.

-t *input-type*

    Select the input data type. Possible values are "ris" (default) and "risx". Other data types have to be converted to one of these types before adding them to the database.

-U *username*

    Provide a different username than that of the current user, so e.g. some technician or administrative staff can add references in behalf of a researcher.

file

    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.

Example

```
refdbc:
checkref -A xhtml -G /usr/local/share/refdb/css/refdb-frequency.css -E ISO-88
```

This command adds the data in `foo.ris` to temporary tables in the current database, using the *ISO-8859-1* encoding. The result of the duplicate checks is requested in *xhtml* format using a stylesheet that displays frequency information graphically. In addition to the default fields the keywords will be listed as well.

deletelink

Synopsis

deletelink [-d *database*] [-h] [[-c *command*] | [-o *outfile*] | [-O *outfile*]] {note-specifier} {link-target...}

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).

Options

-c *command*

    Pipe the output through *command*.

-d *database*

    Specify the database.

`-h`
>   Display a help message explaining the command.

`-o` *filename*
>   Write the output to `filename` instead of to stdout.

`-O` *filename*
>   Append the output to `filename` instead of sending it to stdout.

note-specifier
>   Specify one note by either its :NID: or its :NCK: value.

link-target
>   Specify one or more link targets by means of their :ID:, :CK:, :AU:, :KW:, :JF:, :JO:, :J1:, or :J2: values.

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".

deletenote

Synopsis

`deletenote` [-d *database*] [-h] [[-c *command*] | [-o *outfile*] | [-O *outfile*]] {{*ID*...} | {-f *infile*}}

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.

Caution

It is not possible to delete a note if it belongs to a different user.

Options

`-c` *command*
>   Pipe the output through *command*.

`-d` *database*
>   Specify the database.

`-f` *infile*
>   Read a list of NID values in the RIS format from *infile*.

`-h`
>   Display a help message explaining the command.

`-o` *filename*
>   Write the output to `filename` instead of to stdout.

`-O` *filename*
>   Append the output to `filename` instead of sending it to stdout.

NID
> All other arguments are interpreted as a list of noteID values. Ranges may be used to specify consecutive NIDs. If neither NIDs nor an *infile* are provided, refdbc attempts to read the NIDs from stdin. You can ask refdbc to read NID values from stdin *in addition* to other NIDs by using the `-f stdin` option.

Example

```
refdbc:
deletenote 3 5-10 26
```

This command will delete the extended notes with the ID values 3, 5 through 10, and 26.

deleteref

Synopsis

```
deleteref [-d database] [-h] [[-c command] | [-o outfile] | [-O outfile]] {{ ID ...}
| {-f infile}}
```

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.

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.

Options

`-c command`
> Pipe the output through *command*.

`-d database`
> Specify the database.

`-f infile`
> Read a list of ID values in the RIS format from *infile*. This list may be the result of a previous getref command.

`-h`
> Display a help message explaining the command.

`-o filename`
> Write the output to `filename` instead of to stdout.

`-O filename`
> Append the output to `filename` instead of sending it to stdout.

ID

> All other arguments are interpreted as a list of ID values. Ranges may be used to specify consecutive IDs. If neither IDs nor an *infile* are provided, refdbc attempts to read the IDs from stdin. You can ask refdbc to read ID values from stdin *in addition* to other IDs by using the `-f stdin` option.

Example

```
refdbc:
deleteref 3 5-10 26
```

This command will delete the references with the ID values 3, 5 through 10, and 26.

dumpref

Synopsis

```
dumpref [-b listname] [-d database] [-h] [[-c command] | [-o outfile] | [-O outfile]] {[ ID ...] | [-f file]...}
```

Description

Removes references from the specified personal reference list in the current database. If no personal reference list is specified, the default list (carrying the same name as your database username) will be used instead.

Options

<code>-b listname</code>

> Use the personal reference list named *listname*.

<code>-c command</code>

> Pipe the output through *command*.

<code>-d database</code>

> Specify the database.

<code>-f infile</code>

> Read a list of ID values in the RIS format from *infile*. This list may be the result of a previous getref command.

`-h`

> Display a help message explaining the command.

<code>-o filename</code>

> Write the output to `filename` instead of to stdout.

<code>-O filename</code>

> Append the output to `filename` instead of sending it to stdout.

ID

> All other arguments are interpreted as a list of ID values. Ranges may be used to specify consecutive IDs. If neither IDs nor an *infile* are provided, refdbc attempts to read the IDs from stdin. You can ask refdbc to read ID values from stdin *in addition* to other IDs by using the `-f stdin` option.

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.

getau, geted, getas, getax

Synopsis

getau [-d *database*] [-h] [[-c *command*] | [-o *outfile*] | [-O *outfile*]] [-N *limit[:offset]*][-s *format*]{ *unix-regexp* }

geted [-d *database*] [-h] [[-c *command*] | [-o *outfile*] | [-O *outfile*]] [-N *limit[:offset]*][-s *format*]{ *unix-regexp* }

getas [-d *database*] [-h] [[-c *command*] | [-o *outfile*] | [-O *outfile*]] [-N *limit[:offset]*][-s *format*]{ *regexp* }

getas [-d *database*] [-h] [[-c *command*] | [-o *outfile*] | [-O *outfile*]] [-N *limit[:offset]*][-s *format*]{ *regexp* }

Description

Retrieve all author names that match the regular expression *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. **getax** retrieves authors from any level.

Options

-c *command*
Pipe the output through *command*.

-d *database*
Specify the database.

-h
Display a help message explaining the command.

-N *limit[:offset]*
Limit the number of returned datasets. If *limit* is used all by itself, the first *limit* author names are returned. If the optional *offset* argument is used as well, the first *offset* author names will be skipped, and the next *limit* author names will be returned.

-o *filename*
Write the output to `filename` instead of to stdout.

-O *filename*
Append the output to `filename` instead of sending it to stdout.

-s *format*
Request additional frequency information. "freq" provides the absolute number of references that contain the given author. "relfreq" reports a relative frequency indicator as an integer between 0 and 10.

regexp
All other arguments are interpreted as a unix regular expression which limits the results to matching author names.

Note

Some database engines, like SQLite, do not support Unix-style regular expressions. Use SQL regular expressions instead.

Example

```
refdbc:
getau -o authors.txt '^Simpson'
```

This command will write a list of all authors starting with "Simpson" to the file `authors.txt`.

```
refdbc:
getau -N 5:10
```

This command prints the author names 11 through 15.

getjo, getjf, getj1, getj2

Synopsis

getjo [-a] [-d *database*] [-h] [[-c *command*] | [-o *outfile*] | [-O *outfile*]] [-N *limit[:offset]*] [-s *format*] { *regexp* }

getjf [-a] [-d *database*] [-h] [[-c *command*] | [-o *outfile*] | [-O *outfile*]] [-N *limit[:offset]*] [-s *format*] { *regexp* }

getj1 [-a] [-d *database*] [-h] [[-c *command*] | [-o *outfile*] | [-O *outfile*]] [-N *limit[:offset]*] [-s *format*] { *regexp* }

getj2 [-a] [-d *database*] [-h] [[-c *command*] | [-o *outfile*] | [-O *outfile*]] [-N *limit[:offset]*] [-s *format*] { *regexp* }

Description

Retrieve all journal names that match the regular expression *regexp* in the current database. The *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.

Options

-a
> Return all synonymous journal names, 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**.

-c *command*
> Pipe the output through *command*.

-d *database*
> Specify the database.

-h
> Display a help message explaining the command.

-N *limit[:offset]*

    Limit the number of returned datasets. If *limit* is used all by itself, the first *limit* journal names are returned. If the optional *offset* argument is used as well, the first *offset* journal names will be skipped, and the next *limit* journal names will be returned.

-o *filename*

    Write the output to `filename` instead of to stdout.

-O *filename*

    Append the output to `filename` instead of sending it to stdout.

-s *format*

    Request additional frequency information. "freq" provides the absolute number of references that contain the given journal name. "relfreq" reports a relative frequency indicator as an integer between 0 and 10.

regexp

    All other arguments are interpreted as a unix regular expression which limits the results to matching journal names.

Note

    Some database engines, like SQLite, do not support Unix-style regular expressions. Use SQL regular expressions instead.

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`.

getkw

Synopsis

getkw [-d *database*] [-h] [[-c *command*] | [-o *outfile*] | [-O *outfile*]] [-N *limit[:offset]*] [-s *format*] { *regexp* }

Description

Retrieve all keywords that match the regular expression *regexp* in the current database. If no regexp argument is specified, all keywords in the database will be listed.

Options

-c *command*

    Pipe the output through *command*.

-d *database*

    Specify the database.

-h

    Display a help message explaining the command.

-N *limit[:offset]*

    Limit the number of returned datasets. If *limit* is used all by itself, the first *limit* keywords are returned. If the optional *offset* argument is used as well, the first *offset* keywords will be skipped, and the next *limit* keywords will be returned.

-o *filename*
    Write the output to `filename` instead of to stdout.

-O *filename*
    Append the output to `filename` instead of sending it to stdout.

-s *format*
    Request additional frequency information. "freq" provides the absolute number of references that contain the given keyword. "relfreq" reports a relative frequency indicator as an integer between 0 and 10.

regexp
    All other arguments are interpreted as a unix regular expression which limits the results to matching keywords.

Note

> Some database engines, like SQLite, do not support Unix-style regular expressions. Use SQL regular expressions instead.

Examples

```
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`.

```
~#
refdbc -C getkw -s freq -c "sort -r|cut -d ':' -f 2"|less
```

Here we run the **getkw** command from the shell. The result list contains the frequency of each returned keyword. The data are sorted in descending order according to the frequency, and the frequency information itself is stripped off. Of course there is more than one way to arrive here. E.g. you could use the `-c` option to send the data to *stdout* and pipe them through the argument of the `-c` option in the example shown above.

getnote

Synopsis

```
getnote [-d database] [-E encoding] [-h] [[-c command] | [-o outfile] | [-
O outfile]] [-N limit[:offset]] [-P] [-S sort-string] [-t output-type] {[
search-string]|[-f file]}
```

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.

Options

-c *command*
    Pipe the output through *command*.

-d *database*
    Specify the database.

     `-E` *encoding*

          Select the character encoding for the output data if it is different from the database encoding. You can request any encoding that your local libiconv supports (see **man iconv_open** for a list of available encodings).

     `-f` *infile*

          Read the search string from *infile*. This is a simple way to re-run saved queries.

     `-h`

          Display a help message explaining the command.

     `-N` *limit[:offset]*

          Limit the number of returned datasets. If *limit* is used all by itself, the first *limit* extended notes are returned. If the optional *offset* argument is used as well, the first *offset* notes will be skipped, and the next *limit* notes will be returned.

     `-o` *filename*

          Write the output to `filename` instead of to stdout.

     `-O` *filename*

          Append the output to `filename` instead of sending it to stdout.

     `-P`

          Limit the search to the notes which were added by the current user. If this switch is absent, the whole database will be searched.

     `-S` *sort-string*

          Sort the output. Currently you can sort only by *ID* (the default) or by *PY* (publication year).

     `-t` *output-type*

          Select the type of output. Available are "scrn", "html", "xhtml", and "xnote" for a compact format suitable for terminal browsing, HTML, XHTML, or the native XML format, respectively.

     search-string

          The remainder of the arguments is interpreted as a search string. The syntax of the queries is described in the section query language.

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.

getref

Synopsis

getref [-b *listname*] [-d *database*] [-E *encoding*] [-h] [[-c *command*] | [-o *outfile*] | [-O *outfile*]] [-N *limit[:offset]*] [-s *format-string*] [-S *sort-string*] [-t *output-format*] {[*search-string*] | [-f *file*]}

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. See the countref command if you want to know how many references match your current query without actually retrieving a possibly large amount of reference data.

Options

-b *listname*
    Limit the search to the personal reference list named *listname*.

-c *command*
    Pipe the output through *command*.

-d *database*
    Specify the database.

-E *encoding*
    Select the character encoding for the output data if it is different from the database encoding. You can request any encoding that your local libiconv supports (see **man iconv_open** for a list of available encodings).

-f *infile*
    Read the search string from *infile*. This is a simple way to re-run saved queries.

-h
    Display a help message explaining the command.

-N *limit[:offset]*
    Limit the number of returned datasets. If *limit* is used all by itself, the first *limit* matching references are returned. If the optional *offset* argument is used as well, the first *offset* matching references will be skipped, and the next *limit* matching references will be returned.

-o *filename*
    Write the output to `filename` instead of to stdout.

-O *filename*
    Append the output to `filename` instead of sending it to stdout.

-s *format-string*
    Specify additional fields and pseudo-fields (N1, N2/AB, NX, RP, SN, AD, CY, PB, LX, U1 through U5, M1 through M3) that are not displayed by default, except for the RIS and risx output formats. 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**.

-S *sort-string*
    Sort the output. Currently you can sort only by *ID* (the default) or by *PY* (publication year).

-t *output-type*
    Select the type of output. Available are "scrn", "ris", "risx", "html", "xhtml", "db31", "db31x", "db50x", "teix", "tei5x", "mods", and "bibtex" for a compact format suitable for terminal browsing, the native RIS and risx (XML) formats, HTML, XHTML, DocBook SGML, DocBook XML (DTD-based), DocBook XML (schema-based), TEI P4 XML, TEI P5 XML, MODS, or BibTeX format, respectively.

search-string
    The remainder of the arguments is interpreted as a search string. The syntax of the queries is described in the section query language.

Example

```
refdbc:
getref -t ris -o temp.sgml -E ISO-8859-15 ":AU:='& ^Doe ^Jones' AND :KW:=circular\ d
```

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.

help, ?

Synopsis

```
help
```

```
?
```

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).

Example

```
refdbc:
help
```

This will list the available commands.

listdb

Synopsis

```
listdb [-h] [database-regexp]
```

Description

Lists all available databases if no argument is specified. If `database-regexp` is specified, only the databases matching this expression will be listed.

Options

`-h`
  Displays a help message explaining the **listdb** command.

`database-regexp`
  A valid SQL regular expression which limits the output to matching database names.

Example

```
refdbc:
listdb db%
```

This command will list all available databases that start with "db".

liststyle

Synopsis

```
liststyle { style-regexp }
```

Description

Lists all available bibliography styles that match `style-regexp`.If no argument is specified, all available styles will be listed.

Options

`-h`
　　Displays a help message explaining the **listdb** command.

`style-regexp`
　　A valid Unix regular expression which limits the output to matching style names.

> # Note
>
> Some database engines, like SQLite, do not support Unix-style regular expressions. Use SQL regular expressions instead.

Example

```
refdbc:
liststyle ^J.*
```

This will list all bibliography styles that start with a capital "J".

pickref

Synopsis

pickref [-b `listname`] [-d `database`] [-h] [[-c `command`] | [-o `outfile`] | [-O `outfile`]] {[`ID`...]|[-f `file`]}

Description

Adds references to the specified personal reference list in the current database. If no personal reference list is specified, the default list (carrying the same name as your database username) will be used instead.

Options

`-b listname`
　　Use the personal reference list named `listname`.

`-c command`
　　Pipe the output through `command`.

`-d database`
　　Specify the database.

`-f infile`
　　Read a list of ID values in the RIS format from `infile`. This list may be the result of a previous getref command.

`-h`
　　Display a help message explaining the command.

`-o filename`
　　Write the output to `filename` instead of to stdout.

-O *filename*

> Append the output to filename instead of sending it to stdout.

ID

> All other arguments are interpreted as a list of ID values. Ranges may be used to specify consecutive IDs. If neither IDs nor an *infile* are provided, refdbc attempts to read the IDs from stdin. You can ask refdbc to read ID values from stdin *in addition* to other IDs by using the -f stdin option.

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.

countnote

Synopsis

countnote [-b *listname*] [-c *command*] [-d *database*] [-h] [-N *limit[:offset]*] {[*search-string*] | [-f *file*]}

Description

Counts 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. This command is equivalent to the getnote command except that it does not return the matching notes. It just counts them.

Options

-b *listname*

> Limit the search to the personal reference list named listname.

-c *command*

> Pipe the output through command.

-d *database*

> Specify the database.

-f *infile*

> Read the search string from infile. This is a simple way to re-run saved queries.

-h

> Display a help message explaining the command.

-N *limit[:offset]*

> Limit the number of returned datasets. If limit is used all by itself, the first limit matching references are returned. If the optional offset argument is used as well, the first offset matching references will be skipped, and the next limit matching references will be returned.

search-string

> The remainder of the arguments is interpreted as a search string. The syntax of the queries is described in the section query language.

Example

```
refdbc:
```
countnote :KW:~[rR]eview

This command looks for extended notes that are linked to references which contain keywords like "review" or "Review". The command will return the number of matching notes in the result summary.

countref

Synopsis

```
countref [-b listname] [-c command] [-d database] [-h] [-N limit[:offset]] {[
search-string ]|[-f file]}
```

Description

Counts 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. This command is equivalent to the getref command except that it does not return the matching references. It just counts them.

Options

`-b listname`
Limit the search to the personal reference list named `listname`.

`-c command`
Pipe the output through `command`.

`-d database`
Specify the database.

`-f infile`
Read the search string from `infile`. This is a simple way to re-run saved queries.

`-h`
Display a help message explaining the command.

`-N limit[:offset]`
Limit the number of returned datasets. If `limit` is used all by itself, the first `limit` matching references are returned. If the optional `offset` argument is used as well, the first `offset` matching references will be skipped, and the next `limit` matching references will be returned.

`search-string`
The remainder of the arguments is interpreted as a search string. The syntax of the queries is described in the section query language.

Example

```
refdbc:
```
countref ":AU:='& ^Doe ^Jones' AND :KW:=circular\ dichroism"

This command looks for articles with both an author starting with "Doe" and an author starting with "Jones" that have the keyword "circular dichroism". The command will return the number of matching references in the result summary.

selectdb

Synopsis

```
selectdb [-h] { database }
```

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.

Options

-h
Display a help message explaining the command.

database
The name of the database to be selected.

Example

```
refdbc:
selectdb db1
```

This command will make the database db1 the active database. All further queries and operations will affect this database.

set

Synopsis

```
set [-h] [varname] [varvalue]
```

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:"".

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.

Options

-h
Displays a help message explaining the **set** command.

varname
> The name of the variable whose value should be displayed or set.

varvalue
> The new value of the variable to be set.

Example

```
refdbc:
set timeout 90
```

This command will set the timeout to 90 seconds for the current session.

updatejo

Synopsis

```
updatejo [-d database] [-h] [[-c command] | [-o outfile] | [-O outfile]]
{:XY:=name} {:XY:=name...}
```

Description

Updates the list of synonyms of a particular periodical.

Each periodical can have up to four synonymous names. The full name and an official abbreviation (e.g. according to the Index Medicus for biomedical publications) should always be supplied if available. In addition, refdb can store up to two user-defined abbreviations which may serve as shorthands when adding datasets.

Options

-c command
> Pipe the output through *command*.

-d database
> Specify the database.

-h
> Display a help message explaining the command.

-o filename
> Write the output to `filename` instead of to stdout.

-O filename
> Append the output to `filename` instead of sending it to stdout.

:XY:=name
> XY stands for one of JF, JO, J1, or J2 which denote the full name, the abbreviated name, and the user abbreviations 1 and 2, respectively. You have to supply at least two of these items. The first one selects the periodical by one of its existing names in the database. All other items update or add the names as provided.

Example

```
refdbc:
updatejo :JO:="J.Biol.Chem." :JF:="The Journal of Biological Chemistry"
```

Adds (or changes) the full name of the periodical known by its official abbreviation as "J.Biol.Chem." to read "The Journal of Biological Chemistry".

updatenote

Synopsis

```
updatenote [-d database] [-E encoding] [-h] [[-c command]|[-o outfile]|[-O
outfile]][file...]
```

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.

Options

-c `command`
>   Pipe the output through `command`.

-d `database`
>   Specify the database.

-E `encoding`
>   Select the character encoding for the input data if it is different from the default UTF-8.

-h
>   Display a help message explaining the command.

-o `filename`
>   Write the output to `filename` instead of to stdout.

-O `filename`
>   Append the output to `filename` instead of sending it to stdout.

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.

updateref

Synopsis

```
updateref [-d database] [-E encoding] [-h] [[-c command]|[-o outfile]|[-O
outfile]][-t type][-P][-U username][file]
```

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.

Options

`-c command`
> Pipe the output through *command*.

`-d database`
> Specify the database.

`-E encoding`
> Select the character encoding for the input data if it is different from the default setting. 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 not needed and ignored.

`-h`
> Display a help message explaining the command.

`-o filename`
> Write the output to `filename` instead of to stdout.

`-O filename`
> Append the output to `filename` instead of sending it to stdout.

`-t type`
> Select the input data type. Possible values are "ris" (default) and "risx". Other data types have to be converted to one of these types before adding them to the database.

`-P`
> 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".

`-U username`
> Provide a different username than that of the current user, so e.g. some technician or administrative staff can add references in behalf of a researcher.

`file`
> 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.

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.

verbose

Synopsis

```
verbose [-h]
```

Description

Toggles the verbose mode on or off. If the verbose mode is on, the error messages and warnings may
be some more comprehensible.

Options

-h
   Displays a help message explaining the **verbose** command.

Example

```
refdbc:
verbose
```

Depending on the previous value, this command will either turn the verbose mode on or off.

whichdb

Synopsis

```
whichdb [-h]
```

Description

Displays a plethora of information about the currently selected database.

Options

-h
   Displays a help message explaining the **whichdb** command.

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: pgsql
Created: 2003-12-24 22:27:43 UTC
Using refdb version: 0.9.4-pre2
Last modified: 2003-12-24 22:29:05 UTC
```

Files

PREFIX/etc/refdb/refdbcrc
   The global configuration file of refdbc.

`$HOME/.refdbcrc`
    The user configuration file of refdbc.

See also

RefDB (7), *refdbd* (1), *refdba* (1).

RefDB manual (local copy) PREFIX/share/doc/refdb-<version>/refdb-manual/index.html

RefDB manual (web) <http://refdb.sourceforge.net/manual/index.html>

RefDB on the web <http://refdb.sourceforge.net/>

Author

refdbc was written by Markus Hoenicka <markus@mhoenicka.de>.

Name

bib2ris, bib2ris-utf8 — converts bibtex bibliographic data to the RIS format

Synopsis

`bib2ris [-e log-destination] [-h ] [-j ] [-l log-level] [-L log-file] [-q ] [-s separator] [-v ] [-y confdir] file`

`bib2ris-utf8 [-e log-destination] [-h ] [-j ] [-l log-level] [-L log-file] [-q ] [-s separator] [-v ] [-y confdir] file`

Description

bib2ris converts BibTeX bibliography files into RIS files. Latex commands, including non-ASCII characters written as commands, are preserved in the output. Importing the output of the bib2ris utility directly into RefDB is useful only if you use the data exclusively for LaTeX.

bib2ris-utf8 is a variant which converts foreign characters to UTF-8 and strips all other LaTeX commands by means of the refdb_latex2utf8txt (1) tool. The output of bib2ris-utf8 is the preferred format for import into RefDB as it is suitable for both LaTeX and SGML/XML bibliographies.

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 round-trip conversion 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.

The following considerations apply to the data import into RefDB and the data export from RefDB:

1. BibTeX input data that are not written in UTF-8, that use formatting commands like font name, weight, or posture specifications, or that use LaTeX commands to write foreign and special characters should always be converted with bib2ris-utf8.

2. BibTeX output data will have the LaTeX command characters properly escaped. The data will use the default encoding of your reference database unless you specifically request a different encoding with the getref command or with the refdbib tool. Keep in mind that recent LaTeX installations can work with UTF-8 data using the following incantation in the prolog, allowing the easiest support for all kinds of foreign characters:

```
\usepackage[utf8]{inputenc}
```

Options

`-e log-destination`

log-destination 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.

`-j`

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.

`-l log-level`

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 log-file`

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

`-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.

`-s separator`

Specify 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 version and copyright information, then exits.

`-y confdir`

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.

file

If used, this parameter denotes the names of one or more bibtex files. If no file is specified, bib2ris tries to read the data from stdin. Output is always sent to stdout.

Diagnostics

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:

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

32

parse error; includes file access errors

Configuration

bib2ris evaluates the file `bib2risrc` to initialize itself.

Table 14.2. 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.
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

Variable	Default	Comment
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
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".

Data Processing

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 "forcejabbrev" 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.

Files

PREFIX/etc/refdb/bib2risrc
> The global configuration file of bib2ris.

$HOME/.bib2risrc
> The user configuration file of bib2ris.

See also

RefDB (7), *refdb_latex2utf8txt* (1), *db2ris* (1), *en2ris* (1), *marc2ris* (1), *med2ris* (1).

RefDB manual (local copy) PREFIX/share/doc/refdb-<version>/refdb-manual/index.html

RefDB manual (web) <http://refdb.sourceforge.net/manual/index.html>

RefDB on the web <http://refdb.sourceforge.net/>

Author

bib2ris was written by Markus Hoenicka <markus@mhoenicka.de>.

Name

db2ris — converts DocBook bibliographic data to the RIS format

Synopsis

`db2ris` [-a] [-e *log-destination*] [-h] [-L *log-file*] [-o *file*] [-O *file*] [-r *reftype*]
file

Description

db2ris converts DocBook bibliographic data into RIS files. 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.
db2ris is a wrapper script which invokes OpenJade to do the actual transformation.

Options

-a
   Use full first- and othernames if they are present in the DocBook source. By default, all first- and
   othernames will be abbreviated.

-e *log-destination*
   log-destination 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.

-L *log-file*
   Specify the full path to a log file that will receive the log messages. Typically this would be `/
   var/log/refdba`.

-o *file*
   Send output to *file* instead of to stdout. If *file* exists, its contents will be overwritten.

-O *file*
   Send output to *file* instead of to stdout. If *file* exists, the output will be appended.

-r *reftype*
   Set the default RIS reference type. This type will be used if db2ris cannot infer the reference type
   from the *BiblioEntry* element.

file
   The names of one or more DocBook files.

Configuration

db2ris evaluates the file `db2risrc` to initialize itself.

Table 14.3. db2risrc

Variable	Default	Comment
logfile	/var/log/db2ris.log	The full path of a custom log file. This is used only if logdest is set appropriately.
logdest	0	The destination of the log information. 0 = print to stderr; 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 `BiblioEntry` element does not specify the type.

Data Processing

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.

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.

Files

PREFIX/etc/refdb/db2risrc
:   The global configuration file of db2ris.

$HOME/.db2risrc
:   The user configuration file of db2ris.

See also

RefDB (7), *bib2ris* (1), *en2ris* (1), *marc2ris* (1), *med2ris* (1).

RefDB manual (local copy) PREFIX/share/doc/refdb-<version>/refdb-manual/index.html

RefDB manual (web) <http://refdb.sourceforge.net/manual/index.html>

RefDB on the web <http://refdb.sourceforge.net/>

Author

db2ris was written by Markus Hoenicka <markus@mhoenicka.de>.

Name

en2ris — converts EndNote bibliographic data to the RIS format

Synopsis

en2ris [-e *logdest*] [-f *from-encoding*] [-h] [-l *loglevel*] [-L *logfile*] [-o *file*] [-O *file*] [-q] [-t *to-encoding*] [-y *path*]

Description

en2ris fixes the markup errors encountered in EndNote "RIS" output and writes RIS output to stdout.

Options

-e *log-destination*

log-destination 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.

-f *from-encoding*

Select the input character encoding. Supported encodings are platform-dependent and can usually be found in iconv_open(3). If no encodings are specified, *ISO-8859-1* aka Latin-1 is assumed.

-h

Displays help and usage screen, then exits.

-l *log-level*

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 *log-file*

Specify the full path to a log file that will receive the log messages. Typically this would be /var/log/refdba.

-o *file*

Send output to *file*. If *file* exists, its contents will be overwritten.

-O *file*

Send output to *file*. If *file* exists, the output will be appended.

-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 *to-encoding*

Select the output character encoding. Supported encodings are platform-dependent and can usually be found in iconv_open(3). If no encodings are specified, *UTF-8* is assumed.

-y *confdir*

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.

Configuration

en2ris evaluates the file `en2risrc` to initialize itself.

Table 14.4. 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.

Data Processing

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.

Files

PREFIX/etc/refdb/en2risrc
    The global configuration file of en2ris.

$HOME/.en2risrc
    The user configuration file of en2ris.

See also

RefDB (7), *bib2ris* (1), *db2ris* (1), *marc2ris* (1), *med2ris* (1).

RefDB manual (local copy) PREFIX/share/doc/refdb-<version>/refdb-manual/index.html

RefDB manual (web) <http://refdb.sourceforge.net/manual/index.html>

RefDB on the web <http://refdb.sourceforge.net/>

Author

en2ris was written by Markus Hoenicka <markus@mhoenicka.de>.

Name

marc2ris — converts MARC bibliographic data to the RIS format

Synopsis

marc2ris [-e *log-destination*] [-h] [-l *log-level*] [-L *log-file*] [-m] [-o *outfile*]
[-O *outfile*] [-t *input_type*] [-u *t*/*f*] *file*

Description

marc2ris attempts to extract the information useful to RefDB 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.

Options

By default the script reads USMARC data from stdin and sends RIS data to stdout.

-e *log-destination*

log-destination 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.

-l *log-level*

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 *log-file*

Specify the full path to a log file that will receive the log messages. Typically this would be /var/log/refdba.

`-m`
> Switch on additional MARC output. The output data will be the RIS output interspersed with the source MARC data used to generate the output. This is useful to fix conversion errors manually.

`-o file`
> Send output to *file*. If *file* exists, its contents will be overwritten.

`-O file`
> Send output to *file*. If *file* exists, the output will be appended.

`-t input_type`
> Specify the MARC input type. The default is *MARC21*. Other available types are *UNIMARC* and *UKMARC*.

`-u t|f`
> Request Unicode output if set to "t" (this is the default). marc2ris attempts to convert the input data into Unicode (unless the dataset explicitly states that it already uses Unicode). If the conversion does not seem to work, set this to "f" as some MARC variants do not state the character encoding explicitly.

Configuration

marc2ris evaluates the file `marc2risrc` to initialize itself.

Table 14.5. 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 **sed** 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.
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.

Data Processing

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
```

Files

PREFIX/etc/refdb/marc2risrc
    The global configuration file of marc2ris.

$HOME/.marc2risrc
    The user configuration file of marc2ris.

See also

RefDB (7), *bib2ris* (1), *db2ris* (1), *en2ris* (1), *med2ris* (1).

RefDB manual (local copy) PREFIX/share/doc/refdb-<version>/refdb-manual/index.html

RefDB manual (web) <http://refdb.sourceforge.net/manual/index.html>

RefDB on the web <http://refdb.sourceforge.net/>

Author

marc2ris was written by Markus Hoenicka <markus@mhoenicka.de>.

Name

med2ris — converts Medline/Pubmed bibliographic data to the RIS format

Synopsis

`med2ris` [-e *logdest*] [-f *from-encoding*] [-h] [-i] [-l *loglevel*] [-L *logfile*] [-o *file*]
[-O *file*] [-q] [-t *to-encoding*] [-T *type*] [-y *path*]

Description

med2ris 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.

Options

-e *log-destination*

log-destination 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.

-f *from-encoding*

Select the input character encoding. Supported encodings are platform-dependent and can usually be found in iconv_open(3). If no encodings are specified, *ISO-8859-1* aka Latin-1 is assumed.

-h

Displays help and usage screen, then exits.

-i

Output additional information about unknown or unused tags. Mainly useful to debug the conversion if the input format has changed.

-l *log-level*

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 *log-file*

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

-o *file*

Send output to *file*. If *file* exists, its contents will be overwritten.

-O *file*

Send output to *file*. If *file* exists, the output will be appended.

-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 *to-encoding*

> Select the output character encoding. Supported encodings are platform-dependent and can usually be found in iconv_open(3). If no encodings are specified, *ISO-8859-1* aka Latin-1 is assumed.

-T *type*

> Overrides the automatic type detection. Allowed values for *type* are "tag" and "xml" for the tagged Pubmed format and the XML Pubmed format, respectively.

-y *confdir*

> 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.

Configuration

med2ris evaluates the file `med2risrc` to initialize itself.

Table 14.6. 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.

Data Processing

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.

Files

`PREFIX/etc/refdb/med2risrc`
> The global configuration file of med2ris.

`$HOME/.med2risrc`
> The user configuration file of med2ris.

See also

RefDB (7), *bib2ris* (1), *db2ris* (1), *en2ris* (1), *marc2ris* (1).

RefDB manual (local copy) PREFIX/share/doc/refdb-<version>/refdb-manual/index.html

RefDB manual (web) <http://refdb.sourceforge.net/manual/index.html>

RefDB on the web <http://refdb.sourceforge.net/>

Author

med2ris was written by Markus Hoenicka <markus@mhoenicka.de>.

Name

refdb-pubmed — retrieves an XML representation of Pubmed (http://pubmed.org) entries by their Pubmed IDs

Synopsis

`refdb-pubmed` [-d *database*] [-h] [-m *mode*] [-t *type*] [PMID...]

Description

refdb-pubmed retrieves datasets from the Pubmed database using the efetch utility [http://eutils.ncbi.nlm.nih.gov/corehtml/query/static/efetchlit_help.html]. refdb-pubmed reads Pubmed IDs (PMIDs) either as parameters from the command line, or on stdin. Input data should contain PMIDs separated by non-digits (spaces, tabs, newlines, commas,...). All results are written to stdout.

Options

-d *database*
> Specify the Pubmed database to query. Allowed values are *pubmed*, *pmc*, *journals*, or *omim*. Default is pubmed.

-h
> Displays help and usage screen, then exits.

-m *mode*
> Specify the retrieval mode. Allowed values are *xml*, *html*, *text*, and *asn.1*. Default is xml.

-t *type*
> Select the retrieval type. Allowed values are *uilist*, *abstract*, *citation*, *medline*, or *full*. Default is full.

See also

RefDB (7), *med2ris* (1).

RefDB manual (local copy) PREFIX/share/doc/refdb-<version>/refdb-manual/index.html

RefDB manual (web) <http://refdb.sourceforge.net/manual/index.html>

RefDB on the web <http://refdb.sourceforge.net/>

Author

refdb-pubmed was written by Markus Hoenicka <markus@mhoenicka.de>.

Name

refdb_dos2unix — RefDB tool: convert file formats from DOS to Unix

Synopsis

`refdb_dos2unix` [-h] *file1* [*file2*...]

Description

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 as *Reference Manager* or *EndNote* 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 (including refdba and refdbc) on the workstation side. These clients perform the common tasks of a reference database, like adding and searching entries, as well as generating bibliographies.

refdb_dos2unix is a helper utility that converts a file from DOS format (newline = \r\n) to Unix format (newline = \n).

See also

RefDB manual (local copy) PREFIX/share/doc/refdb-<version>/refdb-manual/index.html

RefDB manual (web) <http://refdb.sourceforge.net/manual/index.html>

Author

RefDB was written by Markus Hoenicka <mhoenicka@users.sourceforge.net>.

This manual page was written by David Nebauer <david@nebauer.org> for the Debian project (but may be used by others).

Name

refdb_latex2utf8txt — converts bibtex bibliographic data to UTF-8 plain text

Synopsis

```
refdb_latex2utf8txt
```

Description

refdb_latex2utf8txt is a helper tool which is used by bib2ris-utf8 to clean up the output of bib2ris. It is implemented as a filter which reads from stdin and writes to stdout, without any command-line options.

Diagnostics

The exit code of **refdb_latex2utf8txt** is always 0 (zero).

Data Processing

- The conversion that this tool performs was designed to facilitate the import of reference data from bibtex, through bib2ris (1), to RefDB.

- Foreign and special characters which are encoded as LaTeX commands are converted to their UTF-8 counterparts.

- All other LaTeX command are stripped off.

See also

RefDB (7), *bib2ris* (1).

RefDB manual (local copy) PREFIX/share/doc/refdb-<version>/refdb-manual/index.html

RefDB manual (web) <http://refdb.sourceforge.net/manual/index.html>

RefDB on the web <http://refdb.sourceforge.net/>

Author

refdb_latex2utf8txt was written by Markus Hoenicka <markus@mhoenicka.de> based on a similar script by Peter Csizmadia.

14.2. 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.

14.2.1. 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 ISSN/ISBN (SN), the URL (UR), and the user (U1 through U5) and misc (M1 through M3) fields. In addition, the pseudo-field codes NX (all notes) and LX (all links, including UR and DOI) can be used. To specify several fields, simply concatenate the field codes, as in -s ADPBRP. -s ALL will display all available fields.

14.2.2. 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

14.2.3. 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].

14.2.4. ris

This is identical to the input format. Use it to export references to other reference management systems.

14.2.5. risx

Retrieves the data as an XML document using the risx DTD. Use it to edit references or to create backups of your databases. The default output includes a docline and an internal subset declaring some

common entities. This format works ok for DTD-based processing systems. If you require namespaced output for further processing with schema-based tools, use the -n option to declare a namespace prefix.

14.2.6. mods

This backend retrieves the data as an XML document using the MODS [http://www.loc.gov/standards/mods] schema. This is another useful exchange format for bibliographic data. Use the -n option to retrieve namespaced output with the given namespace prefix.

14.2.7. 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.

14.2.8. 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.

14.2.9. db31x

The output is essentially the same as with the preceeding backend but you'll get a DocBook XML document instead. The default output works for DTD-based processing with DocBook XML versions 4.x. If you need namespaced output for schema-based processing, use the -n option to define a namespace prefix.

14.2.10. db50x

This backend outputs schema-based DocBook V5. The default output declares a default namespace. You can use the -n option to define a namespace prefix which will be used in all elements.

14.2.11. teix

The TEI XML backend formats the query results as a TEI P4 `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.

14.2.12. tei5x

This backend outputs schema-based TEI P5. The default output declares a default namespace. You can use the -n option to define a namespace prefix which will be used in all elements.

14.3. 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.

14.3.1. 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.

14.3.2. 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.

14.3.3. 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].

14.3.4. xnote

This backend encodes the output in xnote [http://refdb.sourceforge.net/xnote/index.html] documents, the same as the input format for extended notes.

14.4. 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.

14.4.1. 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 ("=", "~" and so on). If your value happens to 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:, :TX:

  The title of the reference, of the secondary title, and of the series title, respectively. :TX: performs a search in all title levels.

:AU:, :A2:, :A3:, :AX:

  The name of an author, of a secondary author/editor, and of a series author, respectively. :AX: performs a search in all author levels.

: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.

: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.

Some pseudo-fields help to retrieve the contents of the RIS fields M1 through M3. RefDB provides these pseudo-fields to better identify the contents of these fields. Please be aware that these pseudo-fields apply only to particular reference types. If in doubt, consult the Reference Manager manual [http://www.referencemanager.com/support/docs/ReferenceManager11.pdf] which contains a list of how the M1-M3 fields are used by each reference type.

:DO:
    The digital objects identifier. [http://www.doi.org/index.html]

:TO:
    The type of work.

:AR:
    The area of a map

:OS:
    The operating system of a computer program

:DG:
    The type of degree of a thesis.

:RT:
    Running time of audiovisual data.

:CI:
    International class code of a patent.

:CU:
    US class code of a patent.

:SE:
    The email address of the sender.

:RE:
    The email address of the recipient.

:MT:
    The type of media.

:NV:
    The total number of volumes in a series

:EI:
    The edition of a book.

:CO:
    The type of computer used for a software or data.

:CF:
    The location of a conference

:RN:
    The registry number.

:CL:
    The classification of an audiovisual material.

:SC:
    The section of a magazine or newspaper

:PN:
    The number of a pamphlet.

:CN:
    The number of a chapter.

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 "|" instead of "&".

Note

If an item in the search contains spaces, they have to be escaped by backslashes, as in "Amino\ Acid\ Sequence".

14.4.2. 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.

14.4.3. 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 :P
```

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.

14.5. 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.

14.5.1. 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 "lifoobar". 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 "fooo", but not "fo" or "foooo".

For further information about regular expressions, see the regex chapter in the MySQL documentation [http://www.mysql.com].

14.5.2. 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 15. Tools for bibliographies

This chapter is a reference for **refdbib**, the RefDB bibliography client, and all associated tools which are required for the transformation of documents containing RefDB bibliographies.

Name

refdbib — the bibliography client of RefDB

Synopsis

```
refdbib [-d database] [-D stylespec-directory] [-e log-destination] [-E
encoding][-f stdin][-h][-i IP-address][-l log-level][-L log-file][-m][-N
number][-p port][-q ][-r ][-S style][-t output-type][-T time][-u name][-v ]
[-V ][-w password][-x ][-y confdir] filename
```

Description

refdbib is a command-line client to generate bibliographies with RefDB(7). refdbib reads the contents of *filename*, which contains a list of citations as an XML document according to *citationlistx.dtd*, and sends a bibliography in the requested format to stdout. If no input file is specified, refdbib tries to read the data from stdin. Unless suppressed, it also writes a style specification file (either a DSSSL or an XSLT stylesheet) and a CSS stylesheet for HTML output to your disk.

refdbib is a low-level tool. It is advisable to use one of the wrappers shipped with RefDB. runbib(1) is a shell script which creates the list of citations, runs refdbib on this list, and transforms the document. refdbnd(1) is a Makefile-based system that encapsulates the bibliography generation and document transformation conveniently.

This man page describes only the startup options of refdbib. Please consult the RefDB manual (see below) for a description of the input and output formats, as well as for post-processing instructions that are required for some output types.

Options

-d *database*
: The name of the default database. You can change the database anytime during an interactive session.

-D *stylespec-directory*
: Specify either a full path or . to use the current working directory for the output of the style specification and CSS files. The latter case is what you usually want if you run refdbib from the directory where your LaTeX or SMGL/XML document is stored. This is also the default if you do not specify a directory at all.

-e *log-destination*
: log-destination 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.

-E *encoding*
: Select an output character encoding. If this option is not used, the bibliography data will use the character encoding of the database. See iconv_open(3) for a list of available encodings.

-f *stdin*
: This 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.

-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(1). Instead of the IP address you can also specify the hostname as long as it can be properly resolved by your system.

`-l` *log-level*

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` *log-file*

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

`-m`

This switch turns errors caused by missing references (i.e. cited but not available in the database) into warnings, causing refdbib to return 0 instead of an error code.

`-N` *number*

Use this 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.

`-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.

`-r`

Use this option to request a raw instead of a cooked bibliography. Raw bibliographies are not formatted in any way and are processed with the standard DocBook or TEI stylesheets instead of with the RefDB driver files.

`-S` *style*

Specifies the bibliography style. This controls the formatting of the bibliography and the in-text citations when the document is processed.

`-t` *output-type*

Select the output type. Use *db31* to generate DocBook SGML bibliographies, *db31x* for DocBook XML bibliographies (DTD-based, up to 4.3), *db50x* for Docbook V5 XML bibliographies (schema-based), *teix* for TEI P4 XML bibliographies, *tei5x* for TEI P5 XML bibliographies, *bibtex* for BibTeX bibliographies, and *rtf* for RTF 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 `-t` *tei5x* option creates a directly usable TEI bibliography.

`-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.

-x
>   Send passwords unencrypted.

-y *confdir*
>   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.

Diagnostics

The exit code is 0 if all went fine. It will be 1 if the command returned an error, or if there was a general error condition during startup like a lack of available memory.

Configuration

refdbib evaluates the `refdbibrc` configuration file at startup to initialize itself.

Table 15.1. refdbibrc

Variable	Default	Comment
refdblib	(none)	The path of a directory containing shareable files like DTDs, HTML templates etc.
defaultdb	(none)	The default database. refdbib 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 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 init file is not readable for anyone else. The default setting causes refdbib to ask for your password interactively.

Variable	Default	Comment
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 refdbs 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.
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	db31	The type of output generated. Use *db31* for DocBook SGML bibliographies, *db31x* for DocBook XML bibliographies, *teix* for TEI XML bibliographies, and *bibtex* 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

Variable	Default	Comment
		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.
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.
no_encrypt	f	If set to 't', passwords are transmitted unencrypted. The default is to encrypt passwords.

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 > my
```

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.

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.

```
$~
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
```

Note

The runbib script does exactly this kind of preprocessing automatically.

Files

PREFIX/etc/refdb/refdbibrc
    The global configuration file of refdbib.

$HOME/.refdbibrc
    The user configuration file of refdbib.

See also

RefDB (7), *refdbd* (1), *runbib* (1), *refdbnd* (1), *refdba* (1), *refdbc* (1).

RefDB manual (local copy) PREFIX/share/doc/refdb-<version>/refdb-manual/index.html

RefDB manual (web) <http://refdb.sourceforge.net/manual/index.html>

RefDB on the web <http://refdb.sourceforge.net/>

Author

refdbib was written by Markus Hoenicka <markus@mhoenicka.de>.

Name

refdbnd — creates a skeleton document and a Makefile for RefDB

Synopsis

Interactive mode:

```
refdbnd
```

Non-interactive mode:

```
refdbnd basename doctype pubtype database style encoding [css-file][fo-
driver][html-driver][xhtml-driver]
```

Description

This script should be the first choice for novices for creating new SGML,XML, or RTF documents for use with RefDB(7). If called without arguments, the script runs in a novice-friendly interactive mode and collects a few answers about the new document. Based on these answers it will create a skeleton document as well as 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, XML, or RTF 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
Versions 4.1.2, 4.2, 4.3, 5.0

TEI XML
Versions P4, P5

Rich Text Format (RTF)
Version 1.9 (M$ Word 2007 and earlier, OpenOffice, AbiWord, and most other contemporary word processors)

The skeleton documents created by the script contain the appropriate prolog for the selected document type. If required (SGML, DTD-based XML), an external entity is declared in the internal subset in order to include the bibliography.

Options

basename
The basename of the document, i.e. sans extension. If you want to end up with *foo.pdf*, use *foo*

doctype
Available options are *db31*, *db40*, and *db41* for DocBook SGML; *db41x*, *db42x*, and *db43x* for DocBook XML; *teix* for TEI XML; *rtf* for RTF

pubtype
This is one of *set*, *book*, or *article* for DocBook, and *TEI.2* for TEI. This argument is ignored by the other document types, but must be present nonetheless (pass "foo" or "bar", depending on your mood).

database

  The name of the database that contains your references.

style

  The name of the bibliography style.

encoding

  The character encoding of the document.

css-file

  The name of a CSS file for the HTML output. This option is meaningless for RTF documents and can be left out.

fo-driver

  A custom driver file for fo output. Please consult the manual for the requirements of such a driver. This option is meaningless for RTF documents and can be left out.

html-driver

  A custom driver file for html output. Please consult the manual for the requirements of such a driver. This option is meaningless for RTF documents and can be left out.

xhtml-driver

  A custom driver file for xhtml output. Please consult the manual for the requirements of such a driver. This option is meaningless for RTF documents and can be left out.

Makefile Targets

The Makefile is set up to process your document properly depending on its type. The following targets are available for SGML and XML documents:

html

  Creates a set of HTML files.

pdf

  Creates a PDF file.

ps

  Creates a Postscript file (SGML documents only).

rtf

  Creates a RTF file. The Rich Text Format can be edited by most Word Processors.

  ### Note

  Not all FO processors offer RTF output, so this target may not work for XML documents with your setup.

all

  This is the default if you call **make** without specifying a target. It will build all of the above targets.

htmldist

  Creates a `.tar.gz` archive of the HTML files.

pdfdist

  Creates a `.tar.gz` archive of the PDF file.

ps

  Creates a `.tar.gz` archive of the Postscript file (SGML documents only).

rtf

  Creates a `.tar.gz` archive of the RTF file.

dist
> This will build all of the above archives.

clean
> This will remove all built files, except for the source document using the full citation format.

shortclean
> This will remove all built files, including the source document using the full citation format.

For RTF output there is only a default target which you can invoke by running **make** without an argument, and the "clean" target which removes generated files.

Files

`PREFIX/share/refdb/db/examples/Makefile.template`

`PREFIX/share/refdb/db/examples/Makefile.rtf.template`

The templates used to create the Makefile.

See also

RefDB (7).

RefDB manual (local copy) PREFIX/share/doc/refdb-<version>/refdb-manual/index.html

RefDB manual (web) <http://refdb.sourceforge.net/manual/index.html>

RefDB on the web <http://refdb.sourceforge.net/>

Author

refdbnd was written by Markus Hoenicka <markus@mhoenicka.de>.

Name

runbib — creates RefDB bibliographies

Synopsis

```
runbib [-a fo-driver] [-b html-driver] [-c xhtml-driver] [-d database] [-
E encoding] [-G css-file] [-h ] [-i IP-address] [-I name] [-j jade-args] [-N
number][-r ][-s ][-S style][-t output-type][-u name][-w password]filename
```

Description

This shell script is a wrapper for the bibliography client refdbib(1) to simplify the creation
of bibliographies. For SGML and XML documents, runbib first runs openjade or xsltproc(1),
respectively, to retrieve a list of citations from your source document. Then it runs refdbib(1) 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(1).

Options

-a `fo-driver`

> The full path of a custom XSL driver file for printed output. This file has to import the appropriate
> RefDB fo driver file.

-b `html-driver`

> The full path of a custom XSL driver file for HTML output. This file has to import the appropriate
> RefDB html driver file.

-c `xhtml-driver`

> The full path of a custom XSL driver file for XHTML output. This file has to import the
> appropriate RefDB xhtml driver file.

-d `database`

> The name of the default database.

-E `encoding`

> Select an output character encoding. If this option is not used, the bibliography data will use the
> character encoding of the database. See iconv_open(3) for a list of available encodings.

-G `CSS-file`

> Set the path or the URL of a CSS file to be used by the (x)html output

-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(1). Instead of the IP
> address you can also specify the hostname as long as it can be properly resolved by your system.

-I `name`

> Change the value of an entity declaration to include marked sections. Multiple entries are possible.
> These are passed as `-i` options to Jade/OpenJade.

-j `jade-args`

> Set additional command line options for Jade/OpenJade

-N *number*

Use this 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.

-r

Use this option to request a raw instead of a cooked bibliography. Raw bibliographies are not formatted in any way and are processed with the standard DocBook or TEI stylesheets instead of with the RefDB driver files.

-s

Skip the ID extraction step of runbib. You have to provide an aptly named XML file containing the requested IDs.

-S *style*

Specifies the bibliography style. This controls the formatting of the bibliography and the in-text citations when the document is processed.

-t *output-type*

Select the output type. Use *db31* to generate DocBook SGML bibliographies, *db31x* for DocBook XML bibliographies, *teix* for TEI XML bibliographies, *bibtex* for BibTeX bibliographies, and *rtf* for RTF bibliographies.

-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.

-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.

Configuration

Instead of using the command-line switches, runbib can also be configured by means of the runbib 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`. Keep in mind that the runbib configuration file covers only those options which are not passed to refdbib(1) as that tool has its own configuration file.

Table 15.2. refdbjaderc

Variable	Default	Comment
jade_includearg	none	Change the value of an entity declaration to include marked sections. Multiple entries are possible. These are passed as `-i` options to Jade/OpenJade.
jadeargs	none	Set additional command line options for Jade/OpenJade. Use e.g. "-E 0" to disable the maximum error limit
outformat	db31	Set the default output format. Supported values are db31 (DocBook SGML 3.1 and later), db31x (DocBook XML 3.1 and later), db50x (DocBook XML 5.0 and later), teix (TEI P4), bibtex, and rtf.

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.

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.xsl`
> 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.xsl`
> 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.

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.

RTF Output

refdbib creates one output file with the reference information. If your document is named `foo.rtf`, **rtfcitations** extracts the citations and writes them into `foo.id.xml` which refdbib uses as an input

file. The refdbib output file will then be named `foo.bib.rtf` and can be used by **refdbrtf** as an input file.

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
```

Files

`PREFIX/etc/refdb/refdbibrc`
   The global configuration file of refdbib(1), which does most of the work of runbib.

`$HOME/.refdbibrc`
   The user configuration file of refdbib.

See also

RefDB (7), *refdbd* (1), *refdbib* (1), *refdbnd* (1), *refdba* (1), *refdbc* (1).

RefDB manual (local copy) PREFIX/share/doc/refdb-<version>/refdb-manual/index.html

RefDB manual (web) <http://refdb.sourceforge.net/manual/index.html>

RefDB on the web <http://refdb.sourceforge.net/>

Author

runbib was written by Markus Hoenicka <markus@mhoenicka.de>.

Name

runbib-missing — displays references not found by runbib(1)

Synopsis

runbib-missing [-b *basename*]

runbib-missing [-f *basename.bib.xml*] *basename.id.xml*

runbib-missing [-]

Description

runbib(1) may report that it was unable to locate some or all of the references cited in the document, but it will not tell you which ones were missing. runbib-missing collects a list of the ID values which caused these failures and sends them to stdout.

Options

-b *basename*
> The basename, optionally including a path, of your document. runbib-missing assumes that the ID file is called basename.id.xml, and the bibliography file is called basename.bib.xml. This is always the case if you use tools like runbib(1) and refdbnd(1) to create your bibliographies. Using this option is the simplest and recommended way to run the script.

-f *basename.bib.xml*
> The relative or full path of the bibliography file created by runbib(1).

-h
> Displays help and usage screen, then exits.

basename.id.xml
> The relative or full path of the ID file created by runbib(1). Optionally you can pipe the data into stdin instead of specifying the file as a parameter.

Configuration

runbib-missing does not use any configuration files.

Example

For the least amount of typing, you should run this script from the directory where your document files are stored. Then the following command will diplay the IDs, if any, of all RefDB citations which were not found in the database during the most recent runbib(1) invocation on your document thesis.xml.

```
~$
runbib-missing -b thesis
```

See also

RefDB (7), *refdbib* (1), *refdbnd* (1), *runbib* (1).

RefDB manual (local copy) PREFIX/share/doc/refdb-<version>/refdb-manual/index.html

---

RefDB manual (web) <http://refdb.sourceforge.net/manual/index.html>

RefDB on the web <http://refdb.sourceforge.net/>

Author

runbib-missing was written by Markus Hoenicka <markus@mhoenicka.de>.

Name

refdbjade — transform SGML and XML documents containing RefDB bibliographies

Synopsis

```
refdbjade [-h ][-I name][-j jade-args][-p prefix][-s stylesheet][-t format]
[-v variable[=name]] file
```

Description

refdbjade uses the stylesheet driver file that you created with runbib(1) 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 refdbxml(1) instead.

Options

-h
  Prints a command synopsis on the screen and exits

-I *name*
  Change the value of an entity declaration to include marked sections. Multiple entries are possible. These are passed as -i options to Jade/OpenJade.

-j *jade-args*
  Set additional command line options for Jade/OpenJade

-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(1) (which in turn is called by runbib(1)) 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).

-v *variable[=value]*
  Change the value of a variable in the stylesheet. Multiple entries are possible. These are passed as such to Jade/OpenJade. The syntax "variable" causes "variable" to be set to "#t". The extended syntax "variable[=value]" sets the variable to the given value.

file
  The names of one or more SGML or XML files. Each document will be processed separately.

Configuration

Instead of using the command-line switches, refdbjade can also be configured by means of the refdbjaderc 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 15.3. refdbjaderc

Variable	Default	Comment
jade_includearg	none	Change the value of an entity declaration to include marked sections. Multiple entries are possible. These are passed as `-i` options to Jade/OpenJade.
jade_variable	none	Change the value of a variable in the stylesheet. Multiple entries are possible. These are passed as `-v` options to Jade/OpenJade. The syntax "variable" causes "variable" to be set to "#t". The extended syntax "variable[=value]" sets the variable to the given value.
jadeargs	none	Set additional command line options for Jade/OpenJade. Use e.g. "-E 0" to disable the maximum error limit
outformat	tex	Set the default output format. Supported values are tex, html, rtf, dvi, pdf, ps, tps, and tpdf

Environment

SGML_CATALOG_FILES

This environment variable is consulted to resolve public identifiers in the SGML source documents. It is advisable to keep local copies of the DTD files and add **OVERRIDE YES** to the top of your catalog files. This ensures that the local copies are used and avoids unnecessary network traffic.

Example

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
```

See also

RefDB (7), *refdbib* (1), *runbib* (1), *refdbxml* (1).

RefDB manual (local copy) PREFIX/share/doc/refdb-<version>/refdb-manual/index.html

RefDB manual (web) <http://refdb.sourceforge.net/manual/index.html>

RefDB on the web <http://refdb.sourceforge.net/>

Author

refdbjade was written by Markus Hoenicka <markus@mhoenicka.de>.

Name

refdbxml — transform XML documents containing RefDB bibliographies

Synopsis

`refdbxml` [-c *fop_config_file*] [-f *fo_processor*] [-h] [-p *xslt-processor*] [-s *stylesheet*] [-t *format*] *file*

Description

refdbxml uses the stylesheet driver file that you created with runbib(1) and feeds it to an XSLT processor to transform your DocBook or TEI XML document to one of the supported output formats. If printable output is requested, the intermediate FO output will be further processed by a FO processor.

Options

-c *fop_config_file*
> The path to a custom configuration file for the FO processor FOP [http://xmlgraphics.apache.org/fop/].

-f *fo_processor*
> The name of the FO processor used to transform FO files into printable output. Supported values are *fop* (default), *passivetex*, *xep*, and *xfor*.

-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 *xslt-processor*
> This determines the XSL processor that is to be used. refdbxml currently knows to handle *Xalan*, *XT*, *Saxon*, and *xsltproc*.

-s *stylesheet*
> This selects the stylesheet driver file. This file is generated by refdbib(1) (which in turn is called by runbib(1)) and contains additional formatting information.

-t *format*
> Select an output format with this option. Possible values are *html\fr, rtf*, and *pdf*.

file
> The names of one or more XML files. Each document will be processed separately.

Configuration

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 15.4. refdbxmlrc

Variable	Default	Comment
xslt_processor	xsltproc	The name of the XSLT processor used to transform XML documents to html or

Variable	Default	Comment
		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.
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.

Resolving 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.

Files

/usr/local/etc/refdb/refdbxmlrc
    The global configuration file of refdbxml

$HOME/.refdbxmlrc
    The user configuration file of refdbxml.

/etc/xml/catalog
    The global XML catalog file, used to resolve public identifiers. Please note that the location of this file is system-dependent. Also, some XSLT processors do not support catalogs, and others require additional classes or plugins to do so.

Example

We'll transform our document (which 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
```

See also

RefDB (7), *refdbib* (1), *runbib* (1), *refdbjade* (1).

RefDB manual (local copy) PREFIX/share/doc/refdb-<version>/refdb-manual/index.html

RefDB manual (web) <http://refdb.sourceforge.net/manual/index.html>

RefDB on the web <http://refdb.sourceforge.net/>

Author

refdbxml was written by Markus Hoenicka <markus@mhoenicka.de>.

Name

refdbrtf — build RTF documents with RefDB bibliographies

Synopsis

`refdbrtf [-b bibfile] [-h]`

Description

refdbrtf takes a RTF bibliography file (such as created with refdbib) and inserts the citations and the bibliographic listing into the file the bibliography was created from. The source RTF document is read from stdin, and the compound RTF document is sent to stdout.

Options

`-b bibfile`

The path to the RTF bibliography file.

`-h`

Prints a command synopsis on the screen and exits

Example

We have authored a source document called mypaper.rtf which contains a variety of citations. Then we invoked runbib which provided a matching bibliography file called mypaper.bib.rtf. The following command will combine the two RTF files to a new compound document with resolved references and a bibliographic listing:

```
~$
refdbrtf -b mypaper.bib.rtf < mypaper.rtf > mypaper.refdb.rtf
```

See also

RefDB (7), *refdbib* (1), *runbib* (1).

RefDB manual (local copy) PREFIX/share/doc/refdb-<version>/refdb-manual/index.html

RefDB manual (web) <http://refdb.sourceforge.net/manual/index.html>

RefDB on the web <http://refdb.sourceforge.net/>

Author

refdbrtf was written by Markus Hoenicka <markus@mhoenicka.de>.

Name

refdbxp — converts short and full citation formats in SGML and XML documents containing RefDB citations.

Synopsis

`refdbxp [-h ] [-s ] [-t `*`input-format`*`] `*`refdbxp`*

Description

refdbxp allows the interconversion of the short and full notation of citations in the supported SGML and XML documents of RefDB(7). See the RefDB manual (see below) for information about the two notations. The conversion is round-trip safe, and it supports mixing short and full notation in the same source document. Full-notation citations will use the correct encoding for first and subsequent citations of the same reference.

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.

Options

`-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 are *db31* (DocBook SGML version 3.1 or later), *db31x* (DocBook XML, all versions), and *teix* (TEI XML).

Example

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 | :
```

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.sgr
```

Second, (o)nsgmlnorm 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.

See also

RefDB (7), *refdbnd* (1).

RefDB manual (local copy) PREFIX/share/doc/refdb-<version>/refdb-manual/index.html

RefDB manual (web) <http://refdb.sourceforge.net/manual/index.html>

RefDB on the web <http://refdb.sourceforge.net/>

Author

refdbxp was written by Markus Hoenicka <markus@mhoenicka.de>.

Name

refdb-ms — a console-based style generator for RefDB

Synopsis

```
refdb-ms
```

Description

refdb-ms is a console-based style generator for RefDB.

In short, this utility consists of a loop in which you are asked to:

- select an element from a list of elements allowed in that part of the style,

- enter or select appropriate values for mandatory attributes,

- select optional attributes for that element and enter their values,

- select an element from a list of elements ...

This process ends when you have exhausted the DTD.

Every time you are presented with a list of elements you have the option of selecting help. Doing so will present a screen with a summary of each element on the list. The same thing happens when you are presented with a list of attributes.

When an attribute or element is required by the DTD this utility will select it automatically. For some objects you will not need to provide input and so this utility will create the object(s) automatically. This can be disconcerting at first. Please read the console feedback carefully.

Whenever you are asked to enter or select a value for an attribute or element, you are given a brief prompt explaining the purpose of the element or attribute. After each element is added to the style you are presented with a "progress report" showing the (major) elements. It may take a little while to understand this feedback. As you add elements they will be appended to the report. When you complete a major element, such as a PUBTYPE or AUTHORLIST, however, its subsidiary elements are no longer displayed -- they are "folded". This saves space and makes it easier to quickly understand where in the style you are at that point in time. As you become more familiar with the structure of the style DTD this display will become increasingly useful.

At certain times within PUBTYPE, AUTHORONLY, YEARONLY and INTEXTDEF elements you are able to delete the previous "major" element. "Major" elements are those corresponding to ris fields. These elements may contain other elements. For example, deleting an AUTHORLIST element will delete all the children elements of that author list style. You can view the complete element before deletion occurs.

When you select some elements (in general, those corresponding to ris fields) you will be given an opportunity to copy the most recently entered element of that type. If you choose to copy, all attributes and sub-elements will be copied. This can save you a lot of time if, for example, your author lists will always have the same formatting.

After you have created your style this utility will save it to a disk file. It will also generate a brief summary of it in html format. This summary can be of great help when entering references in your reference database.

This utility will then offer to upload the style to RefDB. There is no foreseeable way in which this operation could damage your reference data, but you use it at your own risk.

Files

RefDB::Makestyle

This module is required by refdb-ms. It is available as a tar.gz archive or a Debian package from the RefDB website <refdb.sourceforge.net>.

See also

RefDB::Makestyle (3).

RefDB manual (local copy) PREFIX/share/doc/refdb-<version>/refdb-manual/index.html

RefDB manual (web) <http://refdb.sourceforge.net/manual/index.html>

RefDB on the web <http://refdb.sourceforge.net/>

Author

refdb-ms was written by David Nebauer <david@nebauer.org>.

This manual page was written by David Nebauer <david@nebauer.org> for the Debian project (but may be used by others).

Name

rtfcitations — Extracts RefDB citations from a text file and writes them into an XML file using the citationlistx.dtd.

Synopsis

```
rtfcitations [-h ]
```

Description

rtfcitations reads a plain text document (such as the word processor format RTF, hence its name) from stdin and scans it for particular character sequences which denote RefDB citations. The citations are converted to an XML document following the citationlistx.dtd and sent to stdout. The input document may be any structured or non-structured text which sticks to the following conventions:

Simple citations
> Enclose the RefDB citation key in double square brackets, as in "[[Miller1999]]".

Multi-head citations
> Enclose each RefDB citation key in square brackets, and surround the entire multi-head citation with another pair of square brackets, as in "[[Miller1999][Doe2001]]".

The resulting XML file can be fed to refdbib to generate a bibliography file containing the cited works.

Options

-h
> Prints a command synopsis on the screen and exits.

Example

The following command extracts all citations from foo.rtf and writes them to foo.id.xml:

```
~$
rtfcitations < foo.rtf > foo.id.xml
```

See also

RefDB (7), *refdbib* (1).

RefDB manual (local copy) PREFIX/share/doc/refdb-<version>/refdb-manual/index.html

RefDB manual (web) <http://refdb.sourceforge.net/manual/index.html>

RefDB on the web <http://refdb.sourceforge.net/>

Author

rtfcitations was written by Markus Hoenicka <markus@mhoenicka.de>.

Part V. Programmer's manual

Table of Contents

Chapter 16. 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.

16.1. 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 "risx" 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.

16.2. 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_typeofwork

This alphanumeric field with a maximum of 255 characters stores tye type of work of some reference types.

refdb_area

This alphanumeric field with a maximum of 255 characters stores the area information of MAP entries.

refdb_ostype

This alphanumeric field with a maximum of 255 characters stores the type of the operating system of a computer program.

refdb_degree

This alphanumeric field with a maximum of 255 characters stores the type of the degree of a THES reference..

refdb_runningtime

This alphanumeric field with a maximum of 255 characters stores the running time of several multimedia reference types.

refdb_classcodeintl

This alphanumeric field with a maximum of 255 characters stores the international class code of PAT references.

refdb_classcodeus

This alphanumeric field with a maximum of 255 characters stores the US class code of PAT references.

refdb_senderemail

This alphanumeric field with a maximum of 255 characters stores the email address of the sender of an ICOMM reference.

refdb_recipientemail

This alphanumeric field with a maximum of 255 characters stores the email address of the recipient of an ICOMM reference.

refdb_mediatype

This alphanumeric field with a maximum of 255 characters stores the type of the medium of a reference.

refdb_numvolumes

This alphanumeric field with a maximum of 255 characters stores the total number of volumes of a monographic item which is part of a limited series.

refdb_edition

This alphanumeric field with a maximum of 255 characters stores the edition of a monographic item.

refdb_computer

This alphanumeric field with a maximum of 255 characters stores the type of computer of a COMP reference.

refdb_conferencelocation

This alphanumeric field with a maximum of 255 characters stores the location of a conference of a CONF entry.

refdb_registrynum

This alphanumeric field with a maximum of 255 characters stores the registry number of JFULL entries.

refdb_classification
>This alphanumeric field with a maximum of 255 characters stores the classification of multimedia entries.

refdb_section
>This alphanumeric field with a maximum of 255 characters stores the section of a NEWS entry.

refdb_pamphletnum
>This alphanumeric field with a maximum of 255 characters stores the number of the pamphlet of a PAMP entry.

refdb_chapternum
>This alphanumeric field with a maximum of 255 characters stores the number of the chapter of a CHAP entry.

16.3. 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 a alphanumerical field (maximum lenght 255 characters). The name has the format Lastname[,(F.|First)[(M.|Middle)[,Suffix]]].

16.4. 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.

16.5. 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.

16.6. The table t_note

This table contains the extended notes.

note_id

> This unique identifier for entries in t_note is automatically generated by RefDB and is stored as a BIGINT value (INTEGER for SQLite).

note_key

> This is a user-supplied unique identifier for entries in t_note. If not supplied by the user, RefDB will create a unique identifier when the entry is added.

note_title

> This alphanumeric field (up to 255 characters) contains the title of the entry.

note_content

> This alphanumeric field of unlimited length contains the text of the note.

note_content_type

> This alphanumeric field (up to 255 characters) contains the type of the data in the note_content field.

note_content_xmllang

> This alphanumeric field (up to 255 characters) contains an identifier of the language of the data in the note_content field.

note_user_id

> This BIGINT (INTEGER for SQLite) variable points to the user_id in t_user which corresponds to the user who created the note.

note_date

> This DATETIME field stores the date the note was added, or any other user-supplied date.

note_share

> This short integer field stores whether the note may be shared with others (value != zero) or not (value = zero).

16.7. 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.

16.8. The table t_link

This table contains URLs.

link_id
> The unique ID of each user is stored as a BIGINT value (INTEGER for SQLite).

link_url
> This alphanumeric field of unlimited length contains the URL.

16.9. The table t_xauthor

This table cross-references the tables 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.

16.10. The table t_xkeyword

This table cross-references the tables 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).

16.11. The table t_xuser

This table cross-references the tables 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.

16.12. The table t_xnote

This table cross-references the tables t_note and t_refdb.

xnote_id

> The unique ID of an t_xnote entry is stored as a BIGINT value (INTEGER for SQLite).

note_id

> This BIGINT value (INTEGER for SQLite) points to the ID of the extended note in the table t_note.

xref_id

> This is the ID of the database item that the note is linked to, stored as a BIGINT value (INTEGER for SQLite).

xnote_type

> This field stores which type of database item (reference, keyword, author, periodical) the note is attached to.

16.13. The table t_xlink

This table cross-references the tables t_link and t_refdb.

xlink_id

> The unique ID of an t_xlink entry is stored as a BIGINT value (INTEGER for SQLite).

link_id

> This BIGINT value (INTEGER for SQLite) points to the ID of the link in the table t_link.

xref_id

> This is the ID of the database item that the note is linked to, stored as a BIGINT value (INTEGER for SQLite).

xlink_type

    This field stores the type of the link (URL, PDF, fulltext, related, or image).

xlink_source

    This field stores whether the link is used in a reference or in a note.

Chapter 17. 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.

17.1. 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

17.2. 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.

17.3. 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.

17.4. 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.

17.5. 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 18. 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.

18.1. 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 `refdb_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).

18.2. 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 19. Using RefDB in your programs

19.1. General thoughts

RefDB provides command-line clients to access the application server refdbd. This makes it easy to automate tasks through shell scripts or Makefiles. RefDB uses both possibilities for its own purposes. The prime example for a shell script is runbib, which transform your XML documents by invoking, among others, the RefDB bibliography client. Makefiles are more appropriate in other contexts. The Makefile created by refdbnd is just one example.

However, if you want more than that, you'll want to access refdbd from your own programs. Need RefDB support for your favourite editor? It is doable, as you can see from the RefDB integration in Emacs [http://refdb.sourceforge.net/emacs.html] and in Vim [http://refdb.sourceforge.net/vim.html]. Want to access RefDB through the web? There is a PHP implementation of a web front-end in the RefDB sources with plenty of room for improvements, and there is also a SRU interface. Want to have a pointy-clicky frontend so you'll no longer have to envy the Mac and Windoze users of EndNote? Well, this is a challenge that no one has tried yet. Before you fire up your favourite IDE and start coding, please consider the following:

Do not even think about fiddling with the SQL database! Most languages offer a simple way to access SQL databases, and the database engines often provide client libraries for a variety of languages. However, RefDB is more than just a thin layer on top of a SQL database. RefDB normalizes data. RefDB takes data apart and reassembles them when needed. RefDB takes care of character encoding conversions. Even worse, the SQL table schemas may change from one RefDB version to the next. Therefore you should never attempt to access RefDB data from your own programs on the SQL level. Instead, you should rely on the interfaces that RefDB offers.

19.2. Interfaces

There are two interfaces available which will be explained below.

19.2.1. Call the C clients from your program

Using system calls is fairly straightforward. Most programming languages provide constructs to run external programs and to send data to their standard input or to read data from their standard output (rarely, however, both at a time). The following commands give an example how to accomplish this in Perl:

```
$output = `refdbc -d $dbname -C whichdb -u $user -w $pass 2>/dev/null`;
$result = `refdbc -d $dbname -C addref -t $type -u $user -w $pass $infile 2>&1`
```

The first call discards the standard error of the command and writes only the standard output to the variable $output. The second call collects both standard error and standard output in a single variable. Your program would have to parse $result in order to separate both. In both cases Perl variables are used to supply database names, file names, or the reference type.

This kind of access works from most programming languages. For an example written in Lisp, see the Emacs frontend [http://refdb.sourceforge.net/emacs.html] for RefDB. Easy as it may be, running the C clients from your program does have some drawbacks:

- As shown in the examples above, the handling of standard output and standard error is less than perfect. You can retrieve either, or both in a single variable. Your program therefore needs the logic

to separate output from error messages. If you need both separately, you can redirect both to separate files and then read in these files, but this is as kludgy as it sounds.

- On most systems you can't send to standard input and read from standard output at the same time (see this Perl script [http://www.cpan.org/scripts/process-handling/STDIN.STDOUT.connection.pipe] how you can still achieve this). This can be circumvented by either writing the input data to a file (most RefDB commands can read data from files) or by capturing the input data in a file via redirection.

- Last but not least, you need to have the C clients installed along with your own program.

19.2.2. Directly talk to refdbd

This is a somewhat more ambitious approach, as you have to implement the client part of the client/server dialog described here. The client/server dialog is a plain-text protocol which is fairly easy to implement in just about any language. The only tricky part is the password encryption, but that can be handed over to a little C program called eenc that is part of the RefDB sources. The RefDB project currently has client libraries for two languages which provide an implementation of the client part of the client/server protocol. The Perl client module provides an object-oriented access to refdbd which you can use in your programs without having to know anything about the client/server protocol. There is also a (rudimentary and currently unmaintained) implementation in Ruby.

To give you an impression of how you can use a client library, look at the following Perl example which does approximately the same as the examples above:

```
use RefDBClient::Client;

# create a Client object. You can create as many as you need
my $client = new RefDBClient::Client;

# set the initial connection parameters
$client->set_conninfo("127.0.0.1", "9734", "markus", "pass", "refdbtest",
            "/home/markus/literature", "/usr/local/share/refdb/css/refdb.css");

# run the whichdb command. We're not interested in the command summary,
# but read only the data proper into $data
$client->refdb_whichdb();
$data = $client->get_data();

# in order to add references from a RIS file, we first read the file contents
# into a Risdata object
my $risdata = new RefDBClient::Risdata;

$risdata->read_ris("testdata/pubmed.ris");

# now we send the data to the server. This time we read both the command
# summary (which the C clients send to stderr) into $summary and the
# data proper into $data
$summary = $clientc->refdb_addref(undef, $risdata, "ris", "ISO-8859-1");
$data = $clientc->get_data();
```

The advantages and disadvantages of using a client library are as follows:

- You get a clean, object-oriented interface

- No fiddling with standard error or standard output

- However, there may be no client library for your favourite language

19.3. Tips and tricks

This is currently a ragbag of things to consider. It will hopefully be more structured at a later time.

19.3.1. Paging output

If you run commands like getref that are likely to return more information than fits on a screen, you can either try to parse the result in your program and then display it chunk-wise. You may have to parse the data anyway for a different reason, so this option may come at no extra cost. However, if you just want to display the data, you can also use the server-side paging which is implemented in all "get*" commands. These commands take an optional limit argument. The general format is *limit[:offset]*. *limit* is the number of datasets to display, and the optional *offset* is the number of references to skip at the beginning. That is, asking refdbd to limit the result set to 5:10 will cause it to ignore the first 10 datasets and send datasets 11 through 15. You can use this in a loop to retrieve the datasets in groups of 5 until no more datasets are available. Server-side paging is preferable in all cases where a query may return lots of data, as you don't have to buffer them locally.

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/re
```

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();
```

Creates a new Risdata object

20.2.1.2. read_ris

```
$data->read_ris($file);
```

$file ;

loads RIS data from a file

$file
>   path of file

20.2.1.3. get_ris

```
$data->get_ris();
```

returns previously loaded RIS data

20.2.2. RefDBClient::Simplelist

20.2.2.1. new

```
new RefDBClient::Simplelist();
```

creates a new Simplelist element

20.2.3. RefDBClient::Enigma

20.2.3.1. new

```
new RefDBClient::Enigma();
```

creates a new Enigma element

20.2.4. RefDBClient::Client

20.2.4.1. new

```
new RefDBClient::Client();
```

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, $timeout);

$server_ip ;
$port_address ;
$username ;
$password ;
$database ;
$pdf_root ;
$css_url ;
$timeout ;
```

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

$timeout
   Timeout in seconds after which a stale connection is taken down

20.2.4.3. get_status

```
$client->get_status();
```

returns the numerical server status

20.2.4.4. get_status_msg

```
$client->get_status_msg();
```

returns the server status message

20.2.4.5. get_data

```
$client->get_data();
```

returns the data of the most recent command

20.2.4.6. get_summary

```
$client->get_summary();
```

returns the summary of the most recent command

20.2.4.7. refdb_addstyle

```
$client->refdb_addstyle($styledata);
```

$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);
```

$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);
```

$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);
```

$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);
```

$words ;

removes reserved words from the main database

$words
> space-separated list of words

20.2.4.12. refdb_confserv

```
$client->refdb_confserv($command);
```

$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);
```

$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);
```

$databasename ;

deletes a reference database

$dbname
> name of the database

20.2.4.15. refdb_deletestyle

```
$client->refdb_deletestyle($stylename_regexp);
```

$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);
```

$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);
```

$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);
```

$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);
```

$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);
```

```
$stylename_regexp ;
```

lists matching citation/bibliography styles

$stylename_regexp
> regular expression describing the style names

20.2.4.21. refdb_viewstat

```
$client->refdb_viewstat();
```

requests version/connection info from the server

20.2.4.22. refdb_scankw

```
$client->refdb_scankw($dbname);
```

```
$dbname ;
```

runs a thorough keyword scan in the given database

$dbname
> name of the reference database

20.2.4.23. refdb_addref

```
$client->refdb_addref($owner, $refdata, $type, $encoding);
```

```
$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 'risx'

$encoding
> character encoding of the input data (only for RIS data)

20.2.4.24. refdb_checkref

```
$client->refdb_checkref($risdata, $type, $encoding, $outtype);
```

```
$risdata ;
$type ;
$encoding ;
$outtype ;
```

Checks new references against the references in the database without adding them permanently

$risdata
> 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)

$outtype
  output data type, must be one of 'scrn' or 'xhtml'

20.2.4.25. refdb_updateref

```
$client->refdb_updateref($owner,  $is_personal,  $risdata, $type,
$encoding);

$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)

20.2.4.26. refdb_deleteref

```
$client->refdb_deleteref($idlist);

$idlist ;
```

deletes references from the database

$idlist
  string specifying the IDs of the references to be deleted

20.2.4.27. refdb_addnote

```
$client->refdb_addnote($owner, $xnotedata);

$owner ;
$xnotedata ;
```

adds notes to the database

$owner
  owner of the note, if different from the current user

$xnotedata
XML data specifying the note

20.2.4.28. refdb_updatenote

```
$client->refdb_updatenote($owner, $is_personal, $xnotedata);

$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

20.2.4.29. refdb_deletenote

```
$client->refdb_deletenote($idlist);

$idlist ;
```

deletes notes from the database

$idlist
string specifying the ID values of the notes to be deleted

20.2.4.30. refdb_addlink

```
$client->refdb_addlink($linkspec);

$linkspec ;
```

links notes to database objects

$linkspec
string specifying the link(s) to be created

20.2.4.31. refdb_deletelink

```
$client->refdb_deletelink($linkspec);

$linkspec ;
```

unlinks notes from database objects

$linkspec
string specifying the link(s) to be deleted

20.2.4.32. refdb_getas

```
$client->refdb_getas($limit_string, $name_regexp);

$limit_string ;
$name_regexp ;
```

retrieves matching series authors

$limit_string
>   Limits the matching datasets. Must be in the form 'limit[:offset]', where limit is the number of returned datasets, and offset the number of datasets to skip.

$name_regexp
>   regular expression describing the names to be retrieved

20.2.4.33. refdb_getau

```
$client->refdb_getau($limit_string, $name_regexp);
```

```
$limit_string ;
$name_regexp ;
```

retrieves matching authors

$limit_string
>   Limits the matching datasets. Must be in the form 'limit[:offset]', where limit is the number of returned datasets, and offset the number of datasets to skip.

$name_regexp
>   regular expression describing the names to be retrieved

20.2.4.34. refdb_geted

```
$client->refdb_geted($limit_string, $name_regexp);
```

```
$limit_string ;
$name_regexp ;
```

retrieves matching editors

$limit_string
>   Limits the matching datasets. Must be in the form 'limit[:offset]', where limit is the number of returned datasets, and offset the number of datasets to skip.

$name_regexp
>   regular expression describing the names to be retrieved

20.2.4.35. refdb_getkw

```
$client->refdb_getkw($limit_string, $keyword_regexp);
```

```
$limit_string ;
$keyword_regexp ;
```

retrieves matching keywords

$limit_string
>   Limits the matching datasets. Must be in the form 'limit[:offset]', where limit is the number of returned datasets, and offset the number of datasets to skip.

$keyword_regexp
>   regular expression describing the keywords to be retrieved

20.2.4.36. refdb_getjf

```
$client->refdb_getjf($is_all, $limit_string, $journal_regexp);
```

```
$is_all ;
$limit_string ;
$journal_regexp ;
```

retrieves matching periodicals (full names)

$is_all
>   set to 't' if all synonymous journal names shall be returned

$limit_string
>   Limits the matching datasets. Must be in the form 'limit[:offset]', where limit is the number of returned datasets, and offset the number of datasets to skip.

$name_regexp
>   regular expression describing the names to be retrieved

20.2.4.37. refdb_getjo

```
$client->refdb_getjo($is_all, $limit_string, $journal_regexp);
```

```
$is_all ;
$limit_string ;
$journal_regexp ;
```

retrieves matching periodical names (abbrev)

$is_all
>   set to 't' if all synonymous journal names shall be returned

$limit_string
>   Limits the matching datasets. Must be in the form 'limit[:offset]', where limit is the number of returned datasets, and offset the number of datasets to skip.

$name_regexp
>   regular expression describing the names to be retrieved

20.2.4.38. refdb_getj1

```
$client->refdb_getj1($is_all, $limit_string, $journal_regexp);
```

```
$is_all ;
$limit_string ;
$journal_regexp ;
```

retrieves matching periodical names (custom abbrev 1)

$is_all
>   set to 't' if all synonymous journal names shall be returned

$limit_string
>   Limits the matching datasets. Must be in the form 'limit[:offset]', where limit is the number of returned datasets, and offset the number of datasets to skip.

$name_regexp
>   regular expression describing the names to be retrieved

20.2.4.39. refdb_getj2

```
$client->refdb_getj2($is_all, $limit_string, $journal_regexp);
```

```
$is_all ;
$limit_string ;
$journal_regexp ;
```

retrieves matching periodical names (custom abbrev 2)

$is_all
>   set to 't' if all synonymous journal names shall be returned

$limit_string
>   Limits the matching datasets. Must be in the form 'limit[:offset]', where limit is the number of returned datasets, and offset the number of datasets to skip.

$name_regexp
>   regular expression describing the names to be retrieved

20.2.4.40. refdb_getref

```
$client->refdb_getref($type,      $format_string,      $sort_string,
$listname, $encoding, $limit_string, $query_string);

$type ;
$format_string ;
$sort_string ;
$listname ;
$encoding ;
$limit_string ;
$query_string ;
```

retrieves references

$type
>   select output format

$format_string
>   specify additional fields to be retrieved

$sort_string
>   specify sorting key

$listname
>   specify a list name if the search is to be confined to a particular personal reference list

$encoding
>   the character encoding for the output data

$limit_string
>   Limits the matching datasets. Must be in the form 'limit[:offset]', where limit is the number of returned datasets, and offset the number of datasets to skip.

$query_string
>   the query that describes the datasets to be retrieved

20.2.4.41. refdb_countref

```
$client->refdb_countref($listname, $limit_string, $query_string);

$listname ;
$limit_string ;
$query_string ;
```

Counts matching references

> specify a list name if the search is to be confined to a particular personal reference list

$limit_string
> Limits the matching datasets. Must be in the form 'limit[:offset]', where limit is the number of returned datasets, and offset the number of datasets to skip.

$query_string
> the query that describes the datasets to be counted

20.2.4.42. refdb_pickref

```
$client->refdb_pickref($idlist, $listname);

$idlist, $listname ;
```

adds references to the users personal reference list

$idlist
> specifies the ID values of the references to be picked

$listname
> the name of the personal list that the references should be added to. If this string is empty, the default personal list is used instead,

20.2.4.43. refdb_dumpref

```
$client->refdb_dumpref($idlist, $listname);

$idlist, $listname ;
```

removes references from personal reference list

$idlist
> specifies the ID values of the references to be dumped

$listname
> the name of the personal list that the references should be added to. If this string is empty, the default personal list is used instead,

20.2.4.44. refdb_getnote

```
$client->refdb_getnote($type,      $format_string,      $sort_string,
$encoding, $limit_string, $query_string);

$type ;
$format_string ;
$sort_string ;
$encoding ;
$limit_string ;
$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

$limit_string
: Limits the matching datasets. Must be in the form 'limit[:offset]', where limit is the number of returned datasets, and offset the number of datasets to skip.

$query_string
: the query that describes the datasets to be retrieved

20.2.4.45. refdb_countnote

```
$client->refdb_countnote($listname, $limit_string, $query_string);

$listname ;
$limit_string ;
$query_string ;
```

Counts matching extended notes

$listname
: specify a list name if the search is to be confined to a particular personal reference list

$limit_string
: Limits the matching datasets. Must be in the form 'limit[:offset]', where limit is the number of returned datasets, and offset the number of datasets to skip.

$query_string
: the query that describes the datasets to be counted

20.2.4.46. refdb_selectdb

```
$client->refdb_selectdb($dbname);

$dbname ;
```

selects an existing database as the current database

$dbname
: name of the reference database

20.2.4.47. refdb_whichdb

```
$client->refdb_whichdb();
```

displays information about the current database

20.2.4.48. refdb_texbib

```
$client->refdb_texbib($style, $cite_data);

$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

20.2.4.49. refdb_dbib

```
$client->refdb_dbib($type, $style, $encoding, $cite_data);

$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/citationlistx/index.html].

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/citestylex/index.html] 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/xnote/xnote.html].

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. The protocol is versioned in order to allow clients and servers to negotiate whether they can fulfil each other's expectations. The protocol described in this document is version 5.

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 it's 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] [-A *in_format*] [-b *listname*] [-d *database*] [-E *encoding*] [-G *URL*] [-H *host*] [-k] [-n] [-N *limit[:offset]*] [-o *number*] [-p] [-P] [-r] [-R *pdfroot*] [-s *format_string*] [-S *sort_string*] [-t *ref_format*] [-u *username*] [-U *name*] [-w *password*] [-W *newuser_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.

-A *in_format*
 The input data format for addref, updateref, and checkref. Currently supported values are "ris" and "risx".

-b *listname*
 Specifies the name of a personal reference list.

-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 *namespace*
 This option passes a namespace prefix to the getref and getnote commands (applies only to XML output)

-N *limit[:offset]*
　The getref command and all other get* commands use this option to limit the range of datasets to return. *limit* denotes the number of references to return, and *offset* is an optional number of references to skip at the beginning.

-o *number*
　This option is used to set a custom starting number for numeric bibliographies.

-p
　This option is used with the addref and updateref commands. If set, refdbd will 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*
　getbib uses this option to pass the bibliography style. Checkref uses this option to request additional fields in the xhtml output.

-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.

-W *newuser_password*
　The adduser command uses this option to set the password of the new user.

23.3. Commented abstract representation of the client/server protocol

This chapter tries to explain the client/server protocol of all currently supported RefDB 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	6\0\0\0\0			the current protocol version, a terminated string of up to 16 characters. The whole string

step	client	server	message type	comment
				will be converted to an integer internally. The current protocol version of RefDB is a compile-time constant which is defined in `src/connect.h`.
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	the error message status, followed by

step	client	server	message type	comment
			unrecoverable error	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
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

step	client	server	message type	comment
				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\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
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

step	client	server	message type	comment
				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, updateref, checkref

The addref and updateref commands send bibliographic data in various formats to the database in order to add or update them, respectively. The related checkref command also sends bibliographic data to the server in order to check for duplicates. 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	000XXref -s ris [options]\0\0\0\0		ok	the ok message status, followed by the terminated addref command sting
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
	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

step	client	server	message type	comment
		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<message>\0\0\0\0 or 413<message>\0\0\0\0	ok	the message status denoting success of the previous add or update action, followed by the current value of the dataset counter
		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
8		403<data>\0\0\0\0	ok	the ok status signalling that a chunk has been added successfully, followed by an optional terminated string containing a result message
9	000		ok	the ok message status
10		000<data>\0\0\0\0	ok	the ok message status, followed by

step	client	server	message type	comment
				a terminated string containing the command result summary
11	000		ok	the ok message status, ending the dialog

And now the slightly different protocol for risx data:

step	client	server	message type	comment
1	000addref -s risx [options]\0\0\0\0		ok	the ok message status, followed by the terminated addref 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\0\0	error	the error status, followed by a terminated, server-generated error message specifying the error
		408<message>\0\0\0\0 or 413<message>\0\0\0\0	ok	the message status denoting success of the previous add or update action, followed by the current value of the dataset counter
		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
8		403<data>\0\0\0\0	ok	the ok status signalling that a chunk has been added successfully, followed by an optional terminated string containing the result message
9	000		ok	the ok message status
10		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
11	000		ok	the ok message status, ending the dialog

23.3.2.4. addstyle

The internal API command **addstyle** corresponds to refdba: addstyle.

step	client	server	message type	comment
1	000addstyle [options] filename \0\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
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\0	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<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 refdba: 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\0		ok	the ok message status, followed by a terminated string containing a list of names
4		000<summary> \0\0\0\0	ok	the ok message status
	205 801		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 refdba: addword and the refdba: deleteword client commands.

step	client	server	message type	comment
1	000addword [options]\0\0\0\0		ok	the ok message status, followed by the terminated addword command sting
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\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
4		000<data>\0\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\0	ok	the ok message status, followed by a terminated string

step	client	server	message type	comment
				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 refdba: confserv.

step	client	server	message type	comment
1	000confserv command [argument]\0\0\0\0		ok	the ok message status, followed by the terminated confserv command sting
2		000<data>\0\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
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

23.3.2.8. createdb

The internal API command **createdb** corresponds to refdba: createdb.

step	client	server	message type	comment
1	000createdb [options]dbname \0\0\0\0		ok	the ok message status, followed by the terminated createdb command sting
2		000<data>\0\0\0\0	ok	the ok message status, followed by a terminated string

step	client	server	message type	comment
				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\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 refdba: 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 refdbc: deletenote.

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
		111 801	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 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 sting containing the number of bytes required to store the ID list.
2		000	ok	the ok message status

step	client	server	message type	comment
		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 {regexp}\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

step	client	server	message type	comment
				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 refdbc: getau, refdbc: geted, refdbc: getas, refdbc: getkw, refdbc: getjo, refdbc: getjf, refdbc: getj1, refdbc: 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
		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\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 cooked SGML and XML bibliographies.

step	client	server	message type	comment
1	000getbib [options]\0\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)

step	client	server	message type	comment
		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
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
		400<message>\0\0\0\0	error	the error status, followed by a

step	client	server	message type	comment
				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\0	ok	the ok message status, followed by the terminated command summary string.
13	000		ok	the ok message status, finishing the dialog

23.3.2.15. getnote, countnote

The internal API command **getnote** corresponds to refdbc: getnote and refdbc: countnote.

step	client	server	message type	comment
1	000getnote [options] <bytes> \0\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\0		ok	the ok message status, followed by a terminated string containing the query
4		404<data>\0\0\0\0	ok	the ok message status denoting a complete dataset, followed by a

step	client	server	message type	comment
				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
		702, or 801	error	the error message status which terminates the session
8	000		ok	the ok message status, ending the dialog

23.3.2.16. getref, countref

The internal API command **getref** corresponds to refdbc: getref and refdbc: countref.

step	client	server	message type	comment
1	000getref [options] <bytes>\0\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 (including the terminal NULL bytes)

step	client	server	message type	comment
2		000	ok	the ok message status
		301	error	the error message status which terminates the session
		801	error	the error message status which terminates the session
3	000<query-string>\0\0\0\0		ok	the ok message status, followed by a terminated string containing the query
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
		702, or 801	error	the error message status which terminates the session

step	client	server	message type	comment
8	000		ok	the ok message status, ending the dialog

23.3.2.17. getrefx

The internal API command **getrefx** is used by the refdbib tool to retrieve raw SGML and XML bibliographies.

step	client	server	message type	comment
1	000getrefx [options]\0\0\0\0		ok	the ok message status, followed by the terminated getbib command string.
2		000	ok	the ok message status
		204, 701, 801	error	the error message status, terminating the dialog
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\0	error	the error status, followed by a terminated, server-generated error message specifying the error
7	000		ok	the ok message status
8		404	ok	the ok status for a successfully transmitted chunk of data
		402	ok	the ok status for the last chunk of data
		234, 801, 702	error	the error message status
9	000		ok	the ok message status
10		000<summary>\0\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

23.3.2.18. getstyle

The internal API command **getstyle** corresponds to refdba: getstyle.

step	client	server	message type	comment
1	000getstyle [options] [regexp]\0\0\0\0		ok	the ok message status, followed by the terminated getstyle 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

step	client	server	message type	comment
3	000		ok	the ok message status, ending the dialog

23.3.2.19. gettexbib

The internal API command **gettexbib** is used by the refdbib tool to retrieve bibtex bibliographies.

step	client	server	message type	comment
1	000gettexbib [options] data-size \0\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
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, 269, 701, 801, or 842	error	the error message status which terminates the session
5	000		ok	the ok message status

step	client	server	message type	comment
6		402<data>\0\0\0\0	ok	the ok message status denoting the last chunk of data (usually a "footer"), see step 4
7	000		ok	the ok message status
8		000<data>\0\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
9	000		ok	the ok message status, ending the dialog

23.3.2.20. listdb

The internal API command **listdb** corresponds to refdbc: listdb.

step	client	server	message type	comment
1	000listdb [regexp]\0\0\0\0		ok	the ok message status, followed by the terminated listdb/selectdb command sting
2		000<data>\0\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\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. liststyle

The internal API command **liststyle** corresponds to refdbc: liststyle.

step	client	server	message type	comment
1	000liststyle [options] [regexp]\0\0\0\0		ok	the ok message status, followed by the terminated liststyle 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

23.3.2.22. listuser

The internal API command **listuser** corresponds to refdba: listuser.

step	client	server	message type	comment
1	000listuser [options] regexp \0\0\0\0		ok	the ok message status, followed by the terminated listuser 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
3	000		ok	the ok message status
4		000<data>\0\0\0\0	ok	the ok message status, followed by

step	client	server	message type	comment
				a terminated string containing the command result summary
5	000		ok	the ok message status, ending the dialog

23.3.2.23. listword

The internal API command **listword** corresponds to refdba: 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
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

23.3.2.24. pickref

The internal API command **pickref** handles the client commands refdbc: pickref and dumpref.

step	client	server	message type	comment
1	000pickref [options] <bytes> \0\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.

step	client	server	message type	comment
2		000	ok	the ok message status
		111 801	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 note IDs
4		000<data>\0\0\0\0	ok	the ok message status, followed by a report about the pick or dump 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
		227, 228, 234	error	the error status, terminating the dialog
7	000		ok	the ok message status, ending the dialog

23.3.2.25. scankw

The internal API command **scankw** corresponds to refdba: scankw.

step	client	server	message type	comment
1	000scankw -d <databasename>\0\0\0\0		ok	the ok message status, followed by the terminated scankw 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

step	client	server	message type	comment
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

23.3.2.26. 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\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\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. 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

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, 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

23.3.2.28. 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
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

23.3.3. 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

261

failed to create citation key

262

failed to create personal list

263

successfully created personal list

264

failed to delete personal list

265

successfully deleted personal list

266

    personal list not owned by current user

267

    personal list does not exist

268

    create t_temp_xdup failed

269

    style not found

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

426

    reference type changed

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
    SEDITORLIST 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

842
failed to load style

999
summary

23.4. Tools for the client-server protocol

Name

eenc — perform password encryption for RefDB clients

Synopsis

eenc [-p *ABC-DE-FG-HI*] *password*

Description

RefDB(7) clients send passwords to refdbd(1) in an encrypted form. Programmers that implement custom clients in languages other than C or Perl may use this program to perform the password encryption if implementing the algorithm in the target language is too cumbersome. The program writes the encoded password to stdout. As the encoding algorithm is symmetrical, you can use this utility to encode plain text passwords as well as to decode encrypted passwords.

The algorithm used by eenc is derived from the ENIGMA algorithm. Security is limited but should be sufficient for the purposes of a reference management software. Please see the RefDB manual (see below) for further information about the client-server protocol.

Options

-p *ABC-DE-FG-HI*
   The slot and rotor position string sent back by refdbd(1) during the first stage of the client-server protocol. ABC denotes which wheel goes into which slot, and the remaining digits encode the positions of the wheels in slots 0 through 2.

password
   The plain-text password when encrypting, or the encoded password when decrypting.

See also

RefDB (7), *refdbd* (1).

RefDB manual (local copy) PREFIX/share/doc/refdb-<version>/refdb-manual/index.html

RefDB manual (web) <http://refdb.sourceforge.net/manual/index.html>

RefDB on the web <http://refdb.sourceforge.net/>

Author

eenc was written by Markus Hoenicka <markus@mhoenicka.de>.

Colophon

This document was written with GNU Emacs [http://www.gnu.org/software/emacs] on FreeBSD [http://www.freebsd.org]. The XML [http://xml.coverpages.org/xml.html] sources use the DocBook XML DTD [http://www.docbook.org/xml/4.2/index.html] version 4.2. The images were created with Inkscape [http://www.inkscape.org/]. The SVG files were then 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 xsltproc [http://xmlsoft.org/XSLT], FOP [http://xml.apache.org/fop/index.html], and the DocBook XSL stylesheets [http://wiki.docbook.org/topic/DocBookXslStylesheets] 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.