

Vmod-Dbrw User Manual

version 2.7, 10 March 2020

Sergey Poznyakoff.

Published by the Free Software Foundation, 51 Franklin Street, Fifth Floor,
Boston, MA 02110-1301 USA

Copyright © 2013-2020 Sergey Poznyakoff

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover, and no Back-Cover texts. A copy of the license is included in the section entitled “GNU Free Documentation License”.

Short Contents

1	Introduction to vmod-dbrw	1
2	Overview	3
3	Configuration	5
4	Writing Queries	9
5	The <code>rewrite</code> Function	13
6	How to Report a Bug	15
A	GNU Free Documentation License	17
	Concept Index	25

Table of Contents

1	Introduction to vmod-dbrw	1
2	Overview	3
3	Configuration	5
3.1	Expansions	7
4	Writing Queries	9
5	The rewrite Function	13
6	How to Report a Bug	15
Appendix A GNU Free Documentation License		
	17
A.1	ADDENDUM: How to use this License for your documents....	24
	Concept Index	25

1 Introduction to vmod-dbrw

Vmod-dbrw is a module for Varnish Cache¹ which implements database-driven rewrite rules. These rules may be similar to `RewriteRule` directives implemented by `mod_rewrite`² module in Apache or to `Redirect` directives of its `mod_alias` module. What distinguishes the vmod-dbrw rules from these, is that they are handled by Varnish, before the request reaches the httpd server, and that they are stored in an SQL database, which makes them easily manageable.

Some web sites implement thousands of rewrite rules. The purpose of this module is to facilitate deploying and handling them.

¹ <https://www.varnish-cache.org>

² <http://httpd.apache.org/docs/current/rewrite/>

2 Overview

Rewrite rules are stored in a MySQL or PostgreSQL database. The `vmod-dbrw` module does not impose any restrictions on its schema. It only needs to know the SQL query which is to be used to retrieve data. This query is supplied to the module, along with the credentials for accessing the database, by calling the `config` function in the `vcl_recv` subroutine of the Varnish configuration file.

Once the module is configured, the `rewrite` function can be called in the appropriate place of the Varnish configuration file. Its argument is a list of variable assignments separated by semicolons, each assignment having the form `name=value`. When called, `rewrite` expands the SQL query registered with the prior call to `config` by replacing each `$name` construct (a *variable reference*) with the corresponding *value* from its argument. Similarly to the shell syntax, the variable reference can also be written as `${name}`. This latter form can be used in contexts where the variable reference is immediately followed by a letter, digit or underscore, to prevent it from being counted as a part of the name. Special syntax is available for substituting default values and invoking built-in functions during the expansion of the query. See [Section 3.1 \[Expansions\], page 7](#), for a detailed description of these.

Having undergone expansions, the query is sent to the database server. If the query returns no records or if an error occurred, `rewrite` returns empty string. In case of error, it also sets the HTTP header `'X-VMOD-DBRW-Error: 1'`. It can be used in VLC code to provide a special handling for such failures.

The returned set of records (if non-empty) is processed depending on the number of fields it contains.

If the returned set has one or two columns, only the first tuple is used and `rewrite` returns the value of its first column.

Otherwise, if the returned set has three or more columns, the regular expression matching is performed. For the purpose of this discussion, let's refer to the columns as follows: *result*, *regexp*, *value* and *flags*. The *flags* column is optional. Any surplus columns are ignored.

For each returned tuple, the *value* column undergoes variable expansion, using the same algorithm as when preparing the query, and the resulting string is matched with the *regexp* column, which is treated as an extended POSIX regular expression. If the value matches the expression, the *result* column is expanded by replacing *backreferences*: each occurrence of `$digit` (where *digit* stands for a decimal digit from '0' through '9') is replaced by the contents of the *digits* parenthesized subexpression in *regexp*. For compatibility with the traditional usage, the `\digit` notation is also allowed. The resulting value is then returned to the caller.

Optional *flags* column is a comma-separated list of flags that control the matching algorithm.

- 'NC' Treat *regexp* as case-insensitive regular expression.
- 'nocase' Treat *regexp* as case-insensitive regular expression.
- 'case' Treat *regexp* as case-sensitive (default).
- 'eq' Use exact string matching.
- 'QSA' Treat the resulting value as URL; append any query string from the original *value* to it.
- 'qsappend'
- 'QSD' Treat the resulting value as URL; discard any query string attached to the original *value*.
- 'qsdiscard'
- 'redirect=*code*' On success, set the 'X-VMOD-DBRW-Status' header to *code*, which must be a valid HTTP status code.
- 'R=*code*' On success, set the 'X-VMOD-DBRW-Status' header to *code*, which must be a valid HTTP status code.
- 'regex' Use regular expression matching. This is the default. This flag is provided for completeness sake, as a counterpart of 'eq'.

If *regexp* or *value* is NULL, strict matching is assumed (see [strict matching], page 3).

If *flags* is NULL, it is ignored.

3 Configuration

`void config (string dbtype, string params, string query)` [function]

This function configures the module and provides it with the data necessary to connect and use the database. It is normally called from the `vcl_recv` subroutine.

Arguments:

dbtype Type of the database to use. Valid values are ‘mysql’ and ‘pgsql’.

params Database connection parameters. This is a list of ‘*name=value*’ assignments separated by semicolons. The *value* part can be any sequence of characters, excepting white space and semicolon. If *value* contains any of these, they either must be escaped by prepending them with a backslash, or the entire *value* must be enclosed in a pair of (single or double) quotes. The following *escape sequences* are allowed for use in *value*:

Sequence	Replaced by
<code>\a</code>	Audible bell character (ASCII 7)
<code>\b</code>	Backspace character (ASCII 8)
<code>\f</code>	Form-feed character (ASCII 12)
<code>\n</code>	Newline character (ASCII 10)
<code>\r</code>	Carriage return character (ASCII 13)
<code>\t</code>	Horizontal tabulation character (ASCII 9)
<code>\v</code>	Vertical tabulation character (ASCII 11)

Table 3.1: Backslash escapes

If a backslash is immediately followed by a symbol not listed in the above table, it is removed and the symbol is reproduced verbatim.

Valid parameters are:

‘`debug=n`’ Set debugging level. Argument is a decimal number.

‘`server=host`’

Name or IP address of the database server to connect to. If not defined, localhost (‘127.0.0.1’) is assumed. For MySQL databases, if *host* begins with a slash, its value is taken to be the full pathname of the local UNIX socket to connect to.

<code>'port=<i>n</i>'</code>	Port number on the <code>'server'</code> to connect to. Default is <code>'3306'</code> for MySQL and <code>5432</code> for Postgres.
<code>'database=<i>name</i>'</code>	The name of the database to use.
<code>'config=<i>filename</i>'</code>	(MySQL-specific) Read database access credentials and other parameters from the MySQL options file <i>filename</i> .
<code>'group=<i>name</i>'</code>	(MySQL-specific) Read credentials from section <i>name</i> of the options file supplied with the <code>config</code> parameter. Default section name is <code>'client'</code> .
<code>'cacert=<i>filename</i>'</code>	Use secure connection to the database server via SSL. The <i>filename</i> argument is a full pathname of the certificate authority file.
<code>'timeout=<i>n</i>'</code>	Sets idle timeout for a single connection. The connection will be closed and opened again if the module is to use it after <i>n</i> or more seconds since its last use. Set <code>'timeout=-1'</code> to disable idle timeout (session will remain open until the SQL server closes it). Set <code>'timeout=0'</code> to close the connection after each request (not recommended). The default depends on the selected SQL backend. For MySQL, it equals the value of the <code>'wait_timeout'</code> global variable. For PostgreSQL, it is <code>'-1'</code> .
<code>'options=<i>string</i>'</code>	(Postgres-specific) Connection options.
<code>'user=<i>name</i>'</code>	Database user name.
<code>'password=<i>string</i>'</code>	Password to access the database.
<i>query</i>	The SQL query to use. It can contain variable references (<code>\$name</code> or <code>\${name}</code>), which will be expanded to the actual value of the <i>name</i> argument to the function <code>rewrite</code> . See Section 3.1 [Expansions] , page 7, for details.

The example below configures `vmod-dbrw` to use MySQL database `'rewrite'`, with the user name `'varnish'` and password `'guessme'`.

```
import dbrw;

sub vcl_recv {
    dbrw.config("mysql",
        "database=rewrite;user=varnish;password=guesme",
        {"SELECT dest
         FROM redirects
         WHERE host='$host'
         AND url='$url'"});
}
```

3.1 Expansions

The ‘query’ argument to the `dbrw.config` function normally contains variable references. A variable reference has the form ‘`$variable`’ or ‘`${variable}`’, where *variable* is the variable name. When the `dbrw.rewrite` function (see [Chapter 5 \[Rewrite\], page 13](#)) is called, each such reference is expanded to the actual value of *variable* passed in the argument to that function.

The two forms are entirely equivalent. The form with curly braces is normally used if the variable name is immediately followed by an alphanumeric symbol, which will otherwise be considered a part of it. This form also allows for specifying the action to take if the variable is undefined or expands to an empty value.

During variable expansion, the forms below cause `dbrw.rewrite` to test for a variable that is unset or null (i.e., whose value is an empty string). Omitting the colon results in a test only for a variable that is unset.

`${variable:-word}`

Use Default Values. If *variable* is unset or null, the expansion of *word* is substituted. Otherwise, the value of *variable* is substituted.

`${variable:=word}`

Assign Default Values. If *variable* is unset or null, the expansion of *word* is assigned to *variable*. The value of *variable* is then substituted.

`${variable:?word}`

Display Error if Null or Unset. If *variable* is null or unset, the expansion of *word* (or a message to that effect if *word* is not present) is output to the current logging channel. Otherwise, the value of *variable* is substituted.

`${variable:+word}`

Use Alternate Value. If *variable* is null or unset, nothing is substituted, otherwise the expansion of *word* is substituted.

After expanding variables, the query undergoes *command expansion*. Syntactically, a command invocation is

`$(cmd args)`

where *cmd* is the command name, and *args* is a list of arguments separated by whitespace. Arguments can in turn contain variable and command references.

During command expansion, each invocation is replaced by the result of the call to function *cmd* with the supplied arguments.

As of version 2.7 of `vmod-dbrw`, only one function is declared:

urlprefixes *uri* [Command]
Expands to comma-separated list of path prefixes contained in *uri*, starting from the longest one (*uri* itself, with eventual query part stripped off). Single `'/'` is not included in the list. Each list item is quoted. The expansion can be used in the `'IN ()'` SQL conditional.

4 Writing Queries

The query supplied to the `config` function depends on the database schema and on the desired kind of matching (e.g. exact vs. wildcard). To ensure the best performance of the module it is important to design the schema and the query so that the database look up be as fast as possible.

Suppose that you plan to use `vmod-dbrw` to implement redirection rules based on strict matching (see [\[strict matching\]](#), page 3).

The simplest database structure for this purpose (assuming MySQL) will be:

```
CREATE TABLE redirects (
  id INT AUTO_INCREMENT,
  host varchar(255) NOT NULL DEFAULT '',
  url varchar(255) NOT NULL DEFAULT '',
  dest varchar(255) DEFAULT NULL,
  PRIMARY KEY (host,url)
);
```

The columns and their purpose are:

id	An integer uniquely identifying the row. It is useful for table management purposes (e.g. deleting the row).
host	Host part of the incoming request.
url	URL part of the incoming request.
dest	Destination URL to redirect to.

The rewrite function looks up a row that has ‘`host`’ and ‘`url`’ matching the incoming request and, if found, returns the value of its ‘`dest`’ column. The corresponding query is:

```
SELECT dest FROM redirects WHERE host='$host' AND url='$url'
```

The variables ‘`host`’ and ‘`url`’ are supposed to contain the actual host and URL parts of the incoming request.

Handling regular expression matches is a bit trickier. Your query should first return such rows that could possibly match the request. Then the `vmod-dbrw` engine will do the rest, by iterating over the returned set and finding the row that actually matches the request. It will iterate over the rows in the order they were returned by the database server, so it might be necessary to sort them by some criterion beforehand.

The following is an example table structure:

```

CREATE TABLE rewrite (
  id INT AUTO_INCREMENT,
  host varchar(255) NOT NULL DEFAULT '',
  url varchar(255) NOT NULL DEFAULT '',
  dest varchar(255) DEFAULT NULL,
  value varchar(255) DEFAULT NULL,
  pattern varchar(255) DEFAULT NULL,
  flags char(64) DEFAULT NULL,
  weight int NOT NULL DEFAULT '0',
  KEY source (host,url)
);

```

The meaning of `id`, `host`, and `dest` is the same as in the previous example. The meaning of `url` is described below. Other columns are (see [\[regex matching\]](#), page 3):

`value` The value to be compared with the pattern.

`pattern` Regular expression to use.

`flags` Optional flags.

`weight` Relative weight of this row in the set. Rows will be sorted by this column, in ascending order.

The simplest way to select candidate rows is by their ‘`host`’ column:

```

SELECT dest,pattern,value,flags
FROM rewrite
WHERE host='$host'
ORDER BY weight

```

One can further abridge the returned set by selecting only those rows whose `url` column is the prefix of the requested URL:

```

SELECT dest,pattern,value,flags
FROM rewrite
WHERE host='$host'
AND LOCATE(url,'$url')==1
ORDER BY weight

```

Furthermore, the `url` column can contain a path prefix, which can be matched using the `IN` conditional:

```

SELECT dest,pattern,value,flags
FROM rewrite
WHERE host='$host'
AND url IN ($(urlprefixes $url))
ORDER BY weight

```

Notice the use of the ‘`$(urlprefixes $url)`’. This invokes the built-in *function* `urlprefixes`, which expands to comma-separated list of properly quoted pathname prefixes, constructed from its argument. For example, if ‘`$url`’ is ‘`/local/user/local?a=1`’, then the expansion of ‘`$(urlprefixes $url)`’ is:


```
'/local/user/local', '/local/user', '/local'
```


5 The `rewrite` Function

`string rewrite (string args)` [function]

This function is the working horse of the module. It rewrites its argument using the database configured in the previous call to `config` and returns the obtained value.

To do so, it performs the following steps:

- Parameter parsing

The `args` parameter must be a list of `name=value` pairs separated by semicolons. The function parses this string and builds a symbol table.

- Variable expansion

Using the symbol table built in the previous stage, each occurrence of `$name` or `${name}` is replaced by the actual value of the variable `name` from the table. Expanding an undefined variable is considered an error.

- Establishing the database connection

Unless the connection has already been established by a prior call to `rewrite`, the function establishes it using the parameters supplied earlier in a call to `config`. If the connection fails, the function returns NULL immediately.

Database connections are persisting and thread-specific. This means that each thread keeps its own connection to the database and attempts to re-establish it if it goes down for some reason.

- Query execution

The query is sent to the server and the resulting set collected from it.

- Result interpretation

The resulting set is interpreted as described in [\[result interpretation\]](#), [page 3](#). This results in a single value being returned to the caller.

Assuming the database structure similar to the one discussed in the previous chapter, the following example illustrates how to use `rewrite` to redirect the incoming request.

```

sub vcl_recv {
    dbrw.config("mysql",
        "database=rewrite;user=varnish;password=gues sme",
        {"SELECT dest
         FROM redirects
         WHERE host='$host'
         AND url='$url'"});
    set req.http.X-Redirect-To =
        dbrw.rewrite("host=" + req.http.Host + ";" +
            "url=" + req.url);
    if (req.http.X-Redirect-To != "") {
        return(synth(301, "Redirect"));
    }
}

```

The ‘synth’ sub must be provided in order to construct redirection responses:

```

import std;

sub vcl_synth {
    if (resp.status == 301) {
        set resp.http.Location = req.http.X-Redirect-To;
        if (req.http.X-VMOD-DBRW-Status != "") {
            set resp.status =
                std.integer(req.http.X-VMOD-DBRW-Status, 301);
        }
        return (deliver);
    }
}

```

The X-VMOD-DBRW-Status header, if set, contains the status code to be returned to the client (see [\[X-VMOD-DBRW-Status\]](#), page 4). Notice the use of the `vmod_std` module to cast it to integer.

If an error occurred during the rewrite, it is recommended to not cache the response. This way the next request will call rewrite again and eventually complete the rewriting. This can be achieved using the following `vcl_backend_response` fragment:

```

sub vcl_backend_response
{
    if (bereq.http.X-VMOD-DBRW-Error == "1") {
        set beresp.uncacheable = true;
        return (deliver);
    }
}

```

6 How to Report a Bug

Email bug reports to gray@gnu.org.

As the purpose of bug reporting is to improve software, please be sure to include a detailed information when reporting a bug. The minimum information needed is:

- Module version you use.
- A description of the bug.
- Conditions under which the bug appears.
- It is often helpful to send the contents of `config.log` file along with your bug report. This file is created after running `./configure` in `vmod-dbrw` source root directory.

Appendix A GNU Free Documentation License

Version 1.2, November 2002

Copyright © 2000,2001,2002 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document *free* in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of “copyleft”, which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The “Document”, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as “you”. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A “Modified Version” of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A “Secondary Section” is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document’s overall subject (or to related matters) and contains nothing that could fall directly within

that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The “Invariant Sections” are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The “Cover Texts” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A “Transparent” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not “Transparent” is called “Opaque”.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The “Title Page” means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, “Title Page” means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

A section “Entitled XYZ” means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as “Acknowledgements”, “Dedications”, “Endorsements”, or “History”.) To “Preserve the Title” of such a section when you modify the Document means that it remains a section “Entitled XYZ” according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document’s license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque

copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If

there is no section Entitled “History” in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the “History” section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled “Acknowledgements” or “Dedications”, Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled “Endorsements”. Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled “Endorsements” or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version’s license notice. These titles must be distinct from any other section titles.

You may add a section Entitled “Endorsements”, provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled “History” in the various original documents, forming one section Entitled “History”; likewise combine any sections Entitled “Acknowledgements”, and any sections Entitled “Dedications”. You must delete all sections Entitled “Endorsements.”

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an “aggregate” if the copyright resulting from the compilation is not used to limit the legal rights of the compilation’s users beyond what the individual works permit. When the Document is included an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire

aggregate, the Document’s Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled “Acknowledgements”, “Dedications”, or “History”, the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License “or any later version” applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

A.1 ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
Copyright (C) year your name.
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.2
or any later version published by the Free Software Foundation;
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.
A copy of the license is included in the section entitled ‘‘GNU
Free Documentation License’’.
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the “with...Texts.” line with this:

```
with the Invariant Sections being list their titles, with
the Front-Cover Texts being list, and with the Back-Cover Texts
being list.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

Concept Index

This is a general index of all issues discussed in this manual

A

append query string 4

B

backslash interpretation 5

C

`cacert` 6

`case` 4

case-insensitive regular expression 3

case-sensitive regular expression 4

certificate authority file, SSL 6

command expansion 8

`config` 5, 6

configuration 5

D

database 6

database engines 3

database name 6

database password 6

database query 6

database server 5

database server port 5

database user name 6

`debug` 5

debugging level 5

discard query string 4

E

`eq` 4

escape sequences 5

exact matching 4

expansion, command 8

expansion, variable 7

expansions 7

F

FDL, GNU Free Documentation License
..... 17

flags 3

G

group 6

I

idle timeout, SQL 6

initialization 5

M

MySQL options file 6

N

name, database 6

`NC` 3

`nocase` 3

O

options 6

options, MySQL 6

options, PostgreSQL 6

P

`password` 6

`port` 5

Q

`qsappend` 4

`QSA` 4

`qsdiscard` 4

`QSD` 4

`query` 6, 9

query string handling 4

R

redirection code 4

`regex` 4

regex matching 3

regular expression matching 4

regular expression, case-insensitive 3

regular expression, case-sensitive 4

result interpretation	3
rewrite	13
rewrite rules	1, 3

S

server	5
SSL connection, MySQL	6
status code	4
strict matching	3
supported databases	3

T

timeout	6
timeout, idle, SQL	6

U

urlprefixes	8
user	6

V

variable expansion	7
Varnish	1
vcl_recv	5, 13
vcl_synth	14
vmod_std	14

X

X-VMOD-DBRW-Error	3
X-VMOD-DBRW-Status	4, 14